Study of the Kinematic and Dynamic Characteristics of a Wormgear Transmission for Helicopter Applications

D.C. Sun and Qin Yuan
State University of New York at Binghamton
Binghamton, New York

DTIC
ELECTE
JAN 2 5 1995
S G D

19950123 084

U.S. ARMY

ARL

RESEARCH LABORATORY
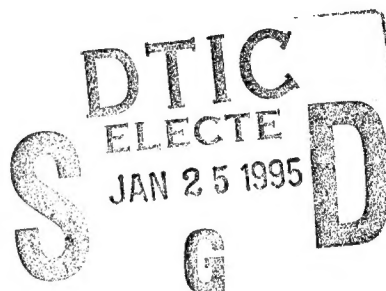
National Aeronautics and
Space Administration

TABLE OF CONTENTS

Abstract

The first phase of the study of the performance of a wormgear transmission is reported. In this phase the work included the selection of a double-enveloping wormgear type, and its dimensions, suitable for use in helicopter transmissions; the 3-D graphics representation of the selected wormgear using the I-DEAS software; the analysis of the kinematics of meshing; the analysis of load sharing among the meshing teeth; and the implementation of the analyses in a computer program. The report describes the analyses, their results, and the use of the computer programs.

Accesion For		
NTIS CRA&I	☒	
DTIC TAB	☐	
Unannounced	☐	
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and / or Special	
A-1		

ii

1   Background and Objectives of the Study

The wormgear is attractive for use in helicopter transmissions because
it can accomplish two functions in one stage, viz. transmitting torque from
the horizontal engine shaft to the vertical rotor shaft, and simultaneously
realizing the required speed ratio.   Thus, the use of the wormgear can result
in compact transmission design with significant weight saving.   However, the
wormgear has a serious drawback.   Because worm and gear teeth are in pure
sliding contact, the sliding friction causes excessive power loss and renders
the efficiency of power transmission low.   Recently, Chaiko (1990) advanced
the concept of using hydrostatic lubrication to remove the sliding friction.
He considered a wormgear transmission of realistic dimensions and under
realistic operating condition, and found that the efficiency of the
transmission could reach 99.14%, and its weight could be only 41% that of the
baseline transmission.   These figures convincingly demonstrate that the
hydrostatically lubricated wormgear can be a candidate of the advanced
helicopter transmissions.   To further investigate the performance of this type
of transmission, an analytical tool capable of generating quantitative data
for a wide range of operating conditions is needed.   Hence a research proposal
was submitted, and a grant was subsequently awarded (NAG3-1316), to develop
such a tool.

The project consists of three years of study with the following
objectives:

1. Year 1 is to be devoted to the selection of a double-enveloping wormgear
    type suitable for use in helicopter transmissions, the analysis of the
    kinematics of meshing, and the analysis of load sharing among the
    meshing teeth.

2. Year 2 is to be devoted to the selection of a hydrostatic lubrication
    configuration on the gear surface, and a comprehensive analysis of the
    hybrid (hydrostatic and hydrodynamic) lubrication at the contact based

1

on the surface geometry and running conditions obtained in Year 1.

3. Year 3 is to be devoted to the analyses of accessory bearings in the wormgear transmission, and the assembly of the obtained information in a computer program package for the design of this type of transmission.

The present report describes the work done in Year 1.

2 Wormgear Geometry

2.1 Selection of Wormgear Type

Wormgears are used to transmit power between two non-parallel non-intersecting shafts. The angle between the shafts is usually 90 deg. There are two common types of wormgears: single-enveloping and double-enveloping. Fig. 2.1 shows a single-enveloping wormgear where the gear envelops the worm. The double-enveloping wormgear is shown in Fig. 2.2, which is like the single-enveloping wormgear except that the worm also envelops the gear.

The double-enveloping wormgear has two contact lines on the mating surface during most of the meshing period. In comparison, the single-enveloping wormgear has only one contact line. Hence, the double-enveloping wormgear is expected to have superior contact properties. The usual way to manufacture a wormgear is first milling a worm, then cutting the gear with a gear hob which has the same shape as the worm. The worm and gear hob are usually ground for high precision applications. The grinding wheel can be a disc (Shen, et al., 1983), Fig. 2.3, or a cone (Simon, 1989), Fig. 2.4. The worm grinding process is called the first enveloping process; and the gear cutting process is called the second enveloping process.

In the first enveloping process the grinding wheel actually rotates around a base circle, with angular velocity $\dot{\psi}_2$, while the worm rotates about

2

its own axis with angular velocity $\dot{\psi}_1$. The distance between the worm axis and the base circle axis is denoted as a_0, and the ratio $\dot{\psi}_1$ to $\dot{\psi}_2$ is denoted as i_0. In the case of the disc grinding wheel, the grinding plane may be parallel to the base circle axis or inclined to it at an angle β. The center distance a and speed ratio i of the wormgear are actually the respective quantities used in the second enveloping process. If $a=a_0$ and $i=i_0$, the wormgear is said to be of the conventional form; otherwise it is said to be of the modified form.

In helicopter applications the torque load and the speed ratio are high. To accommodate such stringent operating conditions the double-enveloping wormgear was selected. This is not only because the double-enveloping wormgear has a larger contact region but also because its contact pattern is favorable to the establishment of fluid film lubrication. Further, for ease of manufacturing the disc grinding method was chosen. The analysis presented below is based on the conventional form, but it can be easily extended to the modified form.

2.2 Layout of Wormgear Dimensions

Wormgear dimensions are usually determined by following the gear standards (e.g. AGMA, 1965), which were developed based on many years of experience. The procedure typically starts with a known speed ratio and a known center distance. For helicopter applications, the suggested transmission power output is 4000 kW , the speed ratio is 110, and the center distance is about 1500 mm.(Chaiko, 1990) These parameters are beyond the range covered by the AGMA standards. Hence, while the standards were followed as closely as possible, some modifications had to be made. Table 1 lists the given and the calculated dimensions and parameters, along with the formulas used for the calculation.

Common terminologies used to describe the double-enveloping wormgear are

3

shown in Fig. 2.5. Another term that should be introduced is the mid-plane, which refers to the plane passing through the worm axis and perpendicular to the gear axis. The pitch circle of the gear is measured in the mid-plane. The pitch circle of the worm is measured at the center of the worm. In the double-enveloping wormgear the worm circular pitch and the gear circular pitch are equal. The base circle is used to define the worm profile in the mid-plane. The normal pressure angle is always chosen to be 20 deg.

Several of the selected quantities in Table 1 should be explained. A right hand worm was selected. The pitch diameter of the worm d_1 was chosen to be 300 mm because of strength and rigidity considerations. The number of meshing teeth N_p was chosen to be 10 instead of the usual 4 or 5 in order to reduce the load level on the meshing teeth. The choice of $\beta = 5$ deg was a compromise between acquiring a better contact condition and maintaining an acceptable tooth thickness at the top of the worm thread.

It should be noted that for the double-enveloping wormgear a different choice of dimensions can result in very different surface shape and contact pattern. The results shown below are based on the data of Table 1.

To better visualize the selected wormgear, a software named I-DEAS was used to generate its 3-D graphics, Figs. 2.6 to 2.9. Fig. 2.6 shows the engaged wormgear. The worm and gear are shown separately in Figs. 2.7 and 2.8. Fig. 2.9 depicts a close-up view of the gear teeth.

Details of the I-DEAS program that generated these graphics, and the procedure to run the program, will be described in Section 5.1.

3   Contact analysis

3.1   Method of Enveloping

The geometric and kinematic properties of the wormgear were analyzed
with the method of enveloping.   To describe the method, let us consider two
rotating machine components, m and k, which are in meshing.   Let us attach a
coordinate system $S_m(x_m, y_m, z_m)$ to the machine component m; and likewise attach
a coordinate system $S_k(x_k, y_k, z_k)$ to the machine component k.   The two machine
components are rotating with the angular coordinates ψ_m and ψ_k with respect to
some stationary reference frame.   Since the two components are in meshing,
there must be some definite relationship between ψ_m and ψ_k.   Hence only one of
them is independent, and either can serve as the running parameter.   Let ψ
denote this running parameter.   The surface of machine component m can be
expressed as $f(x_m, y_m, z_m)=0$ in S_m.   By coordinate transformation, f=0 becomes a
family of surfaces, $F(x_k, y_k, z_k, \psi)=0$ in S_k.   Then the surface of the machine
component k is the envelope of F=0, viz.

$$F(x_k, y_k, z_k, \psi) = 0 \qquad\qquad (1)$$

$$\partial F/\partial \psi = 0$$

If ψ_k/ψ_m = constant, the surfaces of the two meshing components are said to be
conjugate surfaces (Litvin, 1989).

While the idea was simple, the algebraic manipulations needed to carry
it out were quite clumsy.   Recently, Shen, et al. (1983) reformulated the idea
using a set of operators.   Because many of the operators involving the
transformation of coordinates can be done once for all, the method has become
simple as well as elegant.   In the special case of conjugate surfaces, the
method is even more convenient to use.   This will be demonstrated below.   The
reader is referred to Shen, et al. (1983) for the details of the method.

3.2 Coordinate Systems and Operators

To carry out the analysis, four right hand coordinate systems, S_1, S_2, S_j and S_n, are set up. See Fig. 3.1.

S_j and S_n are stationary systems. $S_j(x_j, y_j, z_j)$ is located at the worm position. x_j is along the center line of the wormgear and points to the gear, and z_j axis is along the worm axis. $S_n(x_n, y_n, z_n)$ is located at the gear position. x_n is along the center line but opposite to x_j, and z_n is along the gear axis.

S_1 and S_2 are moving systems. $S_1(x_1, y_1, z_1)$ rotates with the worm and its z_1 axis always coincides with the z_j axis. The worm rotation angle ψ_1 is the angle between x_1 and x_j. $S_2(x_2, y_2, z_2)$ rotates with the gear and its z_2 axis always coincides with the z_n axis. (In the first enveloping process S_2 turns around the base circle with the grinding wheel.) The rotation angle ψ_2 is the angle between x_2 and x_n. Thus, S_1 is fixed to the worm and S_2 is fixed to the gear; and the worm and gear rotate about S_j and S_n, respectively. Initially S_1 coincides with S_j, and S_2 coincides with S_n.

Let $A_\psi^{(km)}$ be the coordinate transformation matrix from S_m to S_k, where the subscript $(\ )_\psi$ indicates that ψ is the running parameter of the coordinate transformation. If the running parameter is understood, the subscript is dropped to simplify the notation. In addition, if the transformation is between two stationary systems, then no running parameter is involved and there will be no subscript. The coordinates in S_m can be transformed into S_k as:

$$\begin{bmatrix} x_k \\ y_k \\ z_k \\ 1 \end{bmatrix} = A_\psi^{(km)} \begin{bmatrix} x_m \\ y_m \\ z_m \\ 1 \end{bmatrix} \qquad (2)$$

6

and a function in S_m can be transformed into S_k as:

$$A^{(km)} f(x_m, y_m, z_m) = f(A^{(km)} x_m, A^{(km)} y_m, A^{(km)} z_m) \tag{3}$$

The coordinate transformation matrices from S_j to S_1, from S_2 to S_n, and from S_n to S_j are given as:

$$A^{(1j)} = \begin{bmatrix} \cos\psi_1 & \sin\psi_1 & 0 & 0 \\ -\sin\psi_1 & \cos\psi_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$A^{(n2)} = \begin{bmatrix} \cos\psi_2 & -\sin\psi_2 & 0 & 0 \\ \sin\psi_2 & \cos\psi_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$A^{(jn)} = \begin{bmatrix} -1 & 0 & 0 & a_0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

The transformation matrix from S_2 to S_1 may be written as $A^{(12)} = A^{(1j)} A^{(jn)} A^{(n2)}$, or:

$$A^{(12)} = \begin{bmatrix} -\cos\psi_1\cos\psi_2 & \cos\psi_1\sin\psi_2 & -\sin\psi_1 & a_0*\cos\psi_1 \\ \sin\psi_1\cos\psi_2 & -\sin\psi_1\sin\psi_2 & -\cos\psi_1 & -a_0*\sin\psi_1 \\ -\sin\psi_2 & -\cos\psi_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Note that the relation $A^{(mk)} = (A^{(km)})^{-1}$ holds.

In addition to the coordinate transformation matrix, two other operators are defined. One is the differentiation operator:

7

$$D_{\psi} = \frac{\partial}{\partial \psi}$$

$$D_{\psi}^2 = D_{\psi} D_{\psi} = \frac{\partial^2}{\partial \psi^2}$$

(8)

The other is the similar differentiation operator:

$$D_{\psi}^{(1)} = A_{\psi}^{(12)} D_{\psi} A_{\psi}^{(21)}$$

$$D_{\psi}^{(2)} = A_{\psi}^{(21)} D_{\psi} A_{\psi}^{(12)}$$

(9)

In the above formulas, the subscript $(\ )_{\psi}$ may be dropped if no confusion can arise.

Several needed properties of these operators are given below:

$$DA^{(km)} f(x_m, y_m, z_m, \psi)$$

$$= A^{(km)} Df + (A^{(km)} f_{x_m}) DA^{(km)} x_m + (A^{(km)} f_{y_m}) DA^{(km)} y_m + (A^{(km)} f_{z_m}) DA^{(km)} z_m$$

(10)

$$DA^{(ij)} f = A^{(ij)} D^{(j)} f$$

$$A^{(ij)} Df = D^{(i)} A^{(ij)} f$$

(11)

$$D^{(j)} f(x_j, y_j, z_j, \psi) = Df + f_{x_j} D^{(j)} x_j + f_{y_j} D^{(j)} y_j + f_{z_j} D^{(j)} z_j$$

(12)

where

$$f_{x_m} = \frac{\partial f}{\partial x_m}, \quad f_{y_m} = \frac{\partial f}{\partial y_m}, \quad f_{z_m} = \frac{\partial f}{\partial z_m}$$

(13)

In the above formulas, k and m can be 1, 2, j and n; but i and j can only be 1 and 2.

To distinguish between the first and the second enveloping processes, ψ is used as the running parameter for the first enveloping process and θ is used for the second enveloping process. The order of operations with respect to different running parameters may be interchanged, e.g.

$$D_\theta A_\psi^{(km)} = A_\psi^{(km)} D_\theta \tag{14}$$

The derivation of the above and other properties of the operators can be found in Shen, et al. (1983).

It should be noted that different symbols, i_0 and i, are used to denote the respective speed ratios in the first and the second enveloping processes, i.e.

$$i_0 = \frac{\dot{\psi}_1}{\dot{\psi}_2} = \frac{\psi_1}{\psi_2}$$

$$i = \frac{\dot{\theta}_1}{\dot{\theta}_2} = \frac{\theta_1}{\theta_2} \tag{15}$$

As said before, if $i_0 = i$ and $a_0 = a$ the wormgear is of the conventional form; otherwise it is of the modified form.

3.3 Worm Surface

The worm surface is formed by a grinding wheel in the first enveloping process. The surface of the grinding wheel is called the original generating surface. For a disc grinding wheel this original generating surface is a plane. For a cone grinding wheel the original generating surface is a line on the cone.

The original generating surface can be expressed in the S_2 system as $f(x_2, y_2, z_2) = 0$. In the first enveloping process S_2 rotates about z_2 with the rotation angle ψ_2, and S_1 rotates about z_1 with the rotation angle ψ_1. As a result of rotation, the original generating surface forms a one-parameter family in S_1, $F(x_1, y_1, z_1, \psi) = A_\psi^{(12)} f(x_2, y_2, z_2) = 0$. Because $\psi_1/\psi_2 = $ constant, the running parameter ψ can be either ψ_1 or ψ_2. Then, the envelope of the family

9

of surfaces is the worm surface, which is given as F=0 and $D_\psi F$=0.

Considering only the case of the selected wormgear, the original generating surface is a plane in S_2, which is:

$$f(x_2, y_2, z_2) = y_2 - z_2 \tan\beta + r_b = 0 \tag{16}$$

After coordinate transformation into S_1, the one-parameter family of planes becomes:

$$F = A_\psi^{(12)} f = x_1(\cos\psi_1 \sin\psi_2 + \tan\beta \sin\psi_1)$$
$$+ y_1(-\sin\psi_1 \sin\psi_2 + \tan\beta \cos\psi_1) \tag{17}$$
$$- z_1 \cos\psi_2 - (a_0 \sin\psi_2 - r_b) = 0$$

Then, the worm surface may be represented by the following two equations:

$$F = x_1(\cos\psi_1 \sin\psi_2 + \tan\beta \sin\psi_1) + y_1(-\sin\psi_1 \sin\psi_2 + \tan\beta \cos\psi_1)$$
$$- z_1 \cos\psi_2 - (a_0 \sin\psi_2 - r_b) = 0$$
$$D_\psi F = x_1(\cos\psi_1 \cos\psi_2 + i_0 \tan\beta \cos\psi_1 - i_0 \sin\psi_1 \sin\psi_2) \tag{18}$$
$$+ y_1(-\sin\psi_1 \cos\psi_2 - i_0 \tan\beta \sin\psi_1 - i_0 \cos\psi_1 \sin\psi_2) + z_1 \sin\psi_2 - a_0 \cos\psi_2 = 0$$

Using the relation $\psi_1/\psi_2 = i_0$ and eliminating ψ_1 and ψ_2 from Eqs. 18, an explicit equation, which is nonlinear, of the worm surface can be obtained.

For a given value of ψ_1, Eqs. 18 represent the equation of a straight line. Recall that the first equation is the equation of the original generating plane expressed at the position ψ_1. Hence the straight line is on both the generating plane and the enveloping surface. This straight line is the contact line between the worm and the disc grinding wheel at the position ψ_1. It can also be seen that the worm surface is the set of all the straight contact lines. Fig. 3.2 depicts this situation.

According to differential geometry, if the planes in a one-parameter family are not parallel and there is not a point common to all the planes, then the envelope of this family is the tangent surface of its edge curve (Kreyszig, 1959). In Fig. 3.2 the curve labelled Γ is the edge curve. The edge curve is given by these equations:

$$F = x_1(\cos\psi_1\sin\psi_2+\tan\beta\sin\psi_1+y_1(-\sin\psi_1\sin\psi_2+\tan\beta\cos\psi_1-z_1\cos\psi_2-(a_0*\sin\psi_2-r_b) = 0$$

$$D_\psi F = x_1(\cos\psi_1\cos\psi_2+i_0\tan\beta\cos\psi_1-i_0\sin\psi_1\sin\psi_2)$$

$$+ y_1(-\sin\psi_1\cos\psi_2-i_0\tan\beta\sin\psi_1-i_0\cos\psi_1\sin\psi_2)+z_1\sin\psi_2-a_0\cos\psi_2=0$$

$$D_\psi D_\psi F = x_1(i_0^2\cos\psi_1\sin\psi_2+i_0^2\tan\beta\sin\psi_1+2*i_0\sin\psi_1\cos\psi_2+\cos\psi_1\sin\psi_2) \qquad (19)$$

$$+ y_1(-i_0^2\sin\psi_1\sin\psi_2+i_0^2\tan\beta\cos\psi_1+2*i_0\cos\psi_1\cos\psi_2-\sin\psi_1\sin\psi_2)-z_1\cos\psi_2-a_0\sin\psi_2= 0$$

The edge curve divides the envelope surface into two branches, of which one is the worm surface. If the worm thread domain determined by the selected dimensions intersects the edge curve, the surface of the disc grinding wheel would extend over the edge curve. The crossed-over portion of the grinding surface would undercut the worm thread. Undercutting the thread would decrease the length of contact lines, decrease the root thickness of the worm thread, and consequently decrease the load carrying capacity of the thread. Hence, wormgear dimensions should be chosen properly so that the worm thread domain is away from the edge curve.

3.4 Gear Surface

The same procedure applies to the second enveloping process. In this process the gear hob has the same shape as the worm surface. Hence, the generating surface is given by Eqs. 18 in S_1. As the result of rotation, this generating surface forms a one-parameter family of surfaces in S_2. The running parameter for this process can be either θ_1 or θ_2, while ψ becomes a geometric parameter already in existence in the worm surface equation. The procedure to obtain the gear surface may be as follows: By eliminating ψ from Eqs. 18, one obtains an explicit equation of the worm surface. This equation may be written as:

$$F^*(x_1, y_1, z_1) = 0 \qquad (20)$$

The one-parameter family of surfaces in S_2 is obtained by coordinate transformation:

$$f^{(2)}(x_2, y_2, z_2, \theta) = A_\theta^{(21)} \ F^* = 0 \qquad (21)$$

The envelope of the family:

$$\begin{aligned} f^{(2)} &= 0 \\ D_\theta \ f^{(2)} &= 0 \end{aligned} \qquad (22)$$

is then the gear surface.

12

But it is not easy to obtain $F^*=0$. Neither is it necessary. Assume that $\psi=\psi(x_1,y_1,z_1)$ is obtained by solving the second equation of Eqs. 18, $D_\psi F=0$. Substituting it into the first equation of Eqs. 18, $F=0$, yields the explicit worm surface equation

$$F^* = F\left(x_1, y_1, z_1, \psi(x_1, y_1, z_1)\right) = 0 \tag{23}$$

Then, the equation $D_\theta f^{(2)} = D_\theta A_\theta^{(21)} F^* = 0$ can be written as:

$$D_\theta A_\theta^{(21)} F^* = A_\theta^{(21)} D_\theta F^* \tag{24}$$
$$+ (A_\theta^{(21)} F^*_{x_1}) D_\theta A_\theta^{(21)} x_1 + (A_\theta^{(21)} F^*_{y_1}) D_\theta A_\theta^{(21)} y_1 + (A_\theta^{(21)} F^*_{z_1}) D_\theta A_\theta^{(21)} z_1 = 0$$

But from Eq. 23,

$$F^*_{x_1} = F_{x_1} + F_\psi \psi_{x_1}$$
$$F^*_{y_1} = F_{y_1} + F_\psi \psi_{y_1} \tag{25}$$
$$F^*_{z_1} = F_{z_1} + F_\psi \psi_{z_1}$$

Then, since $D_\psi F=0$ and Eq. 23 does not contain θ, Eq. 24 reduces to:

$$D_\theta A_\theta^{(21)} F^* = (A_\theta^{(21)} F_{x_1}) D_\theta A_\theta^{(21)} x_1 + (A_\theta^{(21)} F_{y_1}) D_\theta A_\theta^{(21)} y_1 + (A_\theta^{(21)} F_{z_1}) D_\theta A_\theta^{(21)} z_1 = D_\theta A_\theta^{(21)} F = 0 \tag{26}$$

which, on multiplification by $A_\theta^{(21)} A_\theta^{(12)}$, becomes $A_\theta^{(21)} D_\theta^{(1)} F = 0$. Thus, the gear surface is given by:

$$A_\theta^{(21)} F = 0$$
$$A_\theta^{(21)} D_\psi F = 0 \tag{27}$$
$$A_\theta^{(21)} D_\theta^{(1)} F = 0$$

which can be written as:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{28}$$

where

13

$$a_{11} = -\sin\psi_2\cos\theta_2\cos(\psi_1-\theta_1) - \tan\beta\cos\theta_2\sin(\psi_1-\theta_1) + \cos\psi_2\sin\theta_2$$

$$a_{12} = \sin\psi_2\sin\theta_2\cos(\psi_1-\theta_2) + \tan\beta\sin\theta_2\sin(\psi_1-\theta_1) + \cos\psi_2\cos\theta_2$$

$$a_{13} = \sin\psi_2\sin(\psi_1-\theta_1) - \tan\beta\cos(\psi_1-\theta_1) \tag{29}$$

$$a_{14} = a\,[\sin\psi_2\cos(\psi_1-\theta_1) + \tan\beta\sin(\psi_1-\theta_1)] - a_0\sin\psi_2 + r_b$$

$$a_{21} = \sin\psi_2\cos\theta_2\sin(\psi_1-\theta_1) - \tan\beta\cos\theta_2\cos(\psi_1-\theta_1)$$
$$- [\sin\psi_2\sin\theta_2 + \cos\psi_2\cos\theta_2\cos(\psi_1-\theta_1)]/i_0$$

$$a_{22} = -\sin\psi_2\sin\theta_2\sin(\psi_1-\theta_1) + \tan\beta\sin\theta_2\cos(\psi_1-\theta_1)$$
$$- [\sin\psi_2\cos\theta_2 - \cos\psi_2\sin\theta_2\cos(\psi_1-\theta_1)]/i_0$$

$$a_{23} = \sin\psi_2\cos(\psi_1-\theta_1) + \tan\beta\sin(\psi_1-\theta_1) + [\cos\psi_2\sin(\psi_1-\theta_1)]/i_0$$

$$a_{24} = -a\,[\sin\psi_2\sin(\psi_1-\theta_1) - \tan\beta\cos(\psi_1-\theta_1) - \cos\psi_2\cos(\psi_1-\theta_1)/i_0] - a_0\cos\psi_2/i_0 \tag{30}$$

$$a_{31} = a_{12} - i*[\sin\psi_2\cos\theta_2\sin(\psi_1-\theta_1) - \tan\beta\cos\theta_2\cos(\psi_1-\theta_1)]$$

$$a_{32} = -a_{11} + i*[\sin\psi_2\sin\theta_2\sin(\psi_1-\theta_1) - \tan\beta\sin\theta_2\cos(\psi_1-\theta_1)]$$

$$a_{33} = -i*[\sin\psi_2\cos(\psi_1-\theta_1) + \tan\beta\sin(\psi_1-\theta_1)] \tag{31}$$

$$a_{34} = a*i*[\sin\psi_2\sin(\psi_1-\theta_1) - \tan\beta\cos(\psi_1-\theta_1)]$$

In Eqs. 27 there are three equations and five unknowns. Using the relations $\psi_1/\psi_2 = i_0$ and $\theta_1/\theta_2 = i$ the explicit form of the envelope can be obtained by eliminating ψ and θ. Eqs. 27, along with the relation limiting the gear tooth domain, then provide the shape of the gear tooth.

Consider the case of the modified form. For a given value of θ, Eqs. 27

14

produce a curve. The curve is the contact line between the worm and the gear at this θ position. The curve goes over the gear surface twice as shown in Fig. 3.3. For fixed values of ψ and θ, Eqs. 27 represent a point on the contact line.

Consider the case of the conventional form. For a given θ, if one chooses $\psi=\theta$, the third and the second equations of Eqs. 27 are identical. Then Eqs. 27 reproduce a straight contact line of the first enveloping process. This kind of contact lines is called the first enveloping contact lines (the solid lines in Fig. 3.4). The set of all these straight lines reproduce a part of the original generating plane. For a given θ, let $\theta\neq\psi$, Eqs. 27 give a curved contact line, called the second enveloping contact line (the dashed lines in Fig. 3.4). The set of all the curved contact lines form the curved part of the gear surface. Thus, in the case of the conventional form, the gear surface consists of a planar region and a curved region. For a certain range of θ values these two regions overlap each other. The overlapped region is shown in Fig. 3.4 where the dashed lines and the solid lines cross. This situation is impossible to occur in reality. From a view of the (y_2, z_2) plane it is clear that the curved region is farther away from the solid interior than the planar region. Hence, the curved region is fictitious, which is actually hobbed off during manufacturing. Analytically the principal relative curvature is used to sort out the fictitious part (Shen, et al, 1983). The relative curvature is the difference of curvatures of the worm and gear surfaces at the contact point. Because the gear tooth envelops the worm thread, at the contact point the curvature of the gear tooth must not be greater than that of the worm thread (absolute values). Hence, if the relative curvature is negative, the produced gear surface is not real. The derivation of the expression for the principal relative curvature is given in Appendix 3.

From the above description it can be seen that, in the double-enveloping

15

wormgear, the worm and gear are in line contact and there are usually two contact lines on each pair of the meshing teeth.

3.5   Tooth domains

After the equations of the worm and gear surfaces are obtained, the tooth domains need be specified to determine what portions of these surfaces are the actual worm thread and gear tooth.   Table 1 can be used to set up the following geometrical limitations on the worm and gear surfaces.   The worm and gear tooth domains are shown in Fig. 3.5.   Several new symbols that appear in the following equations are also defined in this figure.

The maximum and minimum radii of the worm thread are:

$$r_{w_{max}} = a - \sqrt{(d_{f_2}/2+c)^2 - z_1^2}$$

$$r_{w_{min}} = a - \sqrt{(d_{a_2}/2+c)^2 - z_1^2}$$

$$r_w = \sqrt{x_1^2 + y_1^2} \tag{32}$$

$$r_{w_{min}} \le r_w \le r_{w_{max}}$$

where r_w is the radius of any point on the worm surface.

The maximum and minimum radii of the gear tooth are:

16

$$R_{G_{max}} = a - \sqrt{(d_1/2)^2 - (b_2/2)^2}$$

$$r_{G_{max}} = \min\{(a+\Delta a) - \sqrt{h_{min}^2 - (z_2/2)^2}, \ R_{G_{max}}\}$$

$$r_{G_{min}} = (a+\Delta a) - \sqrt{h_{max}^2 - (z_2/2)^2}$$

$$r_G = \sqrt{x_2^2 + y_2^2} \tag{33}$$

$$r_{G_{min}} \le r_G \le r_{G_{max}}$$

where r_G is the radius of any point on the gear surface, and

$$h_{max} = a - \sqrt{(d_{f_2}/2)^2 - (b_1/2)^2}$$

$$h_{min} = a - \sqrt{(d_{a_2}/2)^2 - (b_1/2)^2} \tag{34}$$

$$\Delta a = (h_{max} + d_{f_2}/2) - a$$

3.6 Properties Needed for Lubrication Studies

To carry out the lubrication analysis several geometric and kinematic properties are needed. These include the shape of the two contacting surfaces, the relative velocity between the two surfaces at a contact point, and the normal forces acting on the meshing teeth, etc.

3.6.1 Surfaces in Stationary Systems

Eqs. 18 and Eqs. 27 are obtained in the moving coordinate systems, where the worm and the gear appear stationary. To see the moving worm and gear surfaces, these equations must be expressed in the stationary systems. At a given instance, or for a given gear position θ, there are two ways to obtain the worm and gear surfaces in a stationary coordinate system. One way is

17

calculating the coordinates in the moving system first, then transforming these points into the stationary system. The other way is transforming the equations into the stationary system first. In either way the coordinate transformation matrices are specialized to the given position. To distinguish this position from the running parameter θ, it is denoted as θ^*. Thus the worm surface in S_j is given as:

$$A_{\theta \cdot}^{(j1)} F = 0$$

$$A_{\theta \cdot}^{(j1)} D_{\phi} F = 0$$

(35)

the gear surface in S_n is given as:

$$A_{\theta \cdot}^{(n2)} A_{\theta}^{(21)} F = 0$$

$$A_{\theta \cdot}^{(n2)} A_{\theta}^{(21)} D_{\phi} F = 0$$

$$A_{\theta \cdot}^{(n2)} A_{\theta}^{(21)} D_{\theta}^{(1)} F = 0$$

(36)

and the gear surface in S_j is given as:

$$A^{(jn)} A_{\theta \cdot}^{(n2)} A_{\theta}^{(21)} F = 0$$

$$A^{(jn)} A_{\theta \cdot}^{(n2)} A_{\theta}^{(21)} D_{\phi} F = 0$$

$$A^{(jn)} A_{\theta \cdot}^{(n2)} A_{\theta}^{(21)} D_{\theta}^{(1)} F = 0$$

(37)

3.6.2 The Relative Velocity at a Contact Point

In Eqs. 35 to 37 if one sets $\theta = \theta^*$, the contact lines expressed in the stationary systems are obtained for the given θ^* position. Then the relative velocity at a contact point can be calculated, which is the difference of the worm velocity and the gear velocity at this contact point. In S_j the relative velocity is given as:

$$\overline{V_r} = \overline{V_1} - \overline{V_2}$$

$$\overline{V_1} = -y_j\,\omega_1\,\overline{i} + x_j\,\omega_1\,\overline{j} \tag{38}$$

$$\overline{V_2} = -y_n\omega_2(-\overline{i}) + x_n\omega_2(-\overline{k})$$

where x_j, y_j, z_j are the coordinates of the contact point in S_j; x_n, y_n, z_n are the corresponding coordinates in S_n; i, j and k are the unit vectors in S_j.

To facilitate further development, let us introduce the term "center point". A center point is the point of intersection between a tooth surface and the gear pitch circle. In the meshing zone there is a gear center point on every gear tooth. Correspondingly there is a worm center point on the worm coil in meshing with this gear tooth. The worm center point is in general slightly different than the gear center point, unless the center point is a contact point. In the analysis below, the normal and friction forces between a pair of meshing teeth are taken as acting at the worm center point, which for brevity will be referred to simply as the center point.

In case that the center point is a contact point, the relative velocity is:

$$V_r = \sqrt{V_1^2 + V_2^2} = \sqrt{(x_j\omega_1)^2 + (d_2\omega_2/2)^2} \tag{39}$$

$$y_j = 0, \qquad x_n^2 + y_n^2 = (d_2/2)^2$$

3.6.3 Normal Direction of the Worm Surface

Because the worm surface is formed by the original generating plane, the normal direction at a point of the worm surface is the same as the normal direction of the original generating plane. Further, since the worm surface and the original generating plane contact in a straight line, all the points on the contact line have the same normal direction. In S_1 this normal

19

direction is:

$$n_x = \cos\psi_1 \sin\psi_2 + \tan\beta \sin\psi_1$$

$$n_y = -\sin\psi_1 \sin\psi_2 + \tan\beta \cos\psi_1 \qquad (40)$$

$$n_z = -\cos\psi_2$$

Transforming Eqs. 40 into S_2, the normal direction of the gear surface at the contact point can also be obtained. Note that since the normal direction of the worm is outward from the worm body, the normal direction of the gear is then inward to the gear body. The normal direction can be further transformed into a stationary coordinate system. Thus, for the given position θ^*, the normal direction in S_j is:

$$n_{jx} = \cos(\psi_1 - \theta_1^*) \sin\psi_2 + \tan\beta \sin(\psi_1 - \theta_1^*)$$

$$n_{jy} = -\sin(\psi_1 - \theta_1^*) \sin\psi_2 + \tan\beta \cos(\psi_1 - \theta_1^*) \qquad (41)$$

$$n_{jz} = -\cos\psi_2$$

3.6.4 Clearance Function

Once the worm and gear surfaces are found, the clearance between the two mating surfaces can be determined. This information is of vital importance in the analysis of fluid film lubrication. This clearance is defined to be the gap between the two surfaces measured in the direction of the common normal on the contact line. Clearly, as the distance from the contact line increases, the gap so measured becomes less representative of the true distance between the two surfaces. But this is immaterial, because as the gap becomes large, the effects of fluid film lubrication also become negligible. The clearance function can be obtained numerically as follows: (1) Choose the grid system on the gear tooth and obtain the coordinates (x_2, y_2, z_2) of the grid points. (2) Calculate the normal direction at a contact point and obtain the equation

20

of the straight line passing through a grid point and in this normal direction. (3) Find the point of intersection between this straight line and the worm surface. The gap at the grid point is then the distance between the grid point and the point of intersection. It is convenient to carry out this calculation in the lubrication analysis, where the grid system has to be selected first.

3.7 Results and Discussion

Sample results were computed for the selected wormgear. Fig. 3.6 shows the wormgear position at $\theta^*=0$, where ten pairs of teeth are in meshing, beginning from gear tooth No.2 and ending with gear tooth No.11. The contact lines on several of these teeth are shown in Figs. 3.7 as viewed in the (x_n, z_n) plane. In each of these figures the circles are on the gear tooth, whose domain is shown by the solid lines; the stars are on that portion of the worm thread that overlapps the gear tooth. Note also that the unit of the coordinates in these figures is millimeter, and the teeth are shown in their true proportion. On gear tooth No.2, Fig. 3.7(a), the two contact lines are far from the mid-plane ($z_n=0$). These contact lines move toward the mid-plane as the tooth number increases. See Figs. 3.7(a) to 3.7(d). On gear tooth No.10, Fig. 3.7(e), the second enveloping contact line crosses over the first enveloping contact line. The crossed-over part is nonexistent, as is explained previously. On gear tooth No.11, Fig. 3.7(f), the entire second enveloping contact line falls below the first enveloping contact line, and becomes non-existent. Thus, toward the end of the meshing zone, the total length of the contact lines diminishes. From these figures it can also be seen that there are two contact lines on the majority of the teeth in the meshing zone; these contact lines are located rather symmetrically on the two sides of the mid-plane and they run across the tooth face more or less in the direction of the x_2 axis.

21

The total length of the contact lines on each meshing tooth is plotted in Fig. 3.8. The situation corresponding to the position $\theta^*=0$ is shown in Fig. 3.8(a). When the gear rotates one degree, i.e. $\theta^*=1$ deg, tooth No.11 moves out of, while tooth No.1 has not moved into, the meshing zone. Hence there are only nine pairs of teeth in meshing. The distribution of the contact line lengths in this situation is shown in Fig. 3.8(b). When the gear rotates one more degree, i.e. $\theta^*=2$ deg, tooth No.1 has moved into, while tooth No.10 still remains in, the meshing zone. There are again ten pairs of teeth in meshing. The corresponding distribution of the contact line lengths is shown in Fig. 3.8(c). Since the angle between two neighboring teeth is $360/N_2$ $=3.273$ deg, if the gear further rotates 1.27 deg, the situation of Fig. 3.8(a) re-appears.

The variation of the contact line length on one gear tooth as it moves through the meshing zone is shown in Fig. 3.9, which is essentially the three graphs of Figs. 3.8 combined.

The radial position of the edge curve on the worm surface is shown in Fig. 3.10. For comparison, the radial position of the hour-glass-shaped worm pitch cylinder is plotted as the dashed curve. Since the tooth depth is about 20 mm, it can be seen that the edge curve is located far below the worm thread. Hence there is no problem of undercutting the worm thread.

From the above results one can identify the following merits of the selected wormgear:

1. On the majority of the meshing teeth there are two contact lines. This means that the contact region on a tooth of the double-enveloping wormgear is larger than that of the single-enveloping wormgear, which only possesses one contact line on a meshing tooth.

2. The contact lines run across the tooth face more or less in the direction of the x_2 axis. Since the relative velocity of the meshing

22

surfaces is nearly perpendicular to the x_2 axis, the two directions meet at a large angle. This feature is favorable to the establishment of fluid film lubrication at the contact.

3. On the majority of the meshing teeth, the pattern of the contact lines is nearly symmetrical with respect to the center point. This feature allows a major simplification to be made in the load sharing analysis.

4. With the chosen gear dimensions there is no problem of undercutting the worm thread.

4 Load Sharing Analysis

The methodology of determining how the load is distributed among meshing teeth is not well developed. Rational analyses are rare and complicated. For spur and helical gears, the load sharing analyses are usually based on the variation of tooth stiffness along contact lines (Elkholy, 1989). In the practical design of these gears, the approach is to analyze an idealized meshing model, then apply various factors to modify the results. To estimate the magnitudes of these load distribution factors, the common approach is to assume that gear teeth behave like a series of thin slices that deflect independently (Steward, 1989). It is doubtful that these methods are suitable to the double-enveloping wormgear, because its contact pattern is vastly different than those of the spur and helical gears. Hence, for the double-enveloping wormgear, there does not exist a theory of load sharing. Neither are there adequate experimental data to guide the analysis and design. In the following, we shall base our analysis of load sharing on a hypothesis.

4.1 Hypothesis on Load Sharing

At a contact point the forces between the meshing surfaces consist of a normal component, that acts in the direction of the common normal, and a

23

friction component. The latter acts in the direction of the relative velocity and has a magnitude equal to the normal component multiplied by the Coulomb friction coefficient. Since the direction of the normal and that of the relative velocity vary with position, these forces vary from contact point to contact point not only in magnitude but also in direction. However, the contact analysis in Section 3 reveals that the contact line pattern on each tooth is nearly symmetrical with respect to the center point. In addition, because of the large speed ratio (and consequently the small lead angle) of the selected wormgear, the variation of the normal direction on a tooth surface is small. These two features allow the following simplifications to be made:

1. The resultant normal force on a tooth is proportional to the contact line length on the tooth. It acts at the center point and is in the normal direction of the center point.

2. The resultant friction force on a tooth acts at the center point and is in the direction of the relative velocity at the center point.

The above simplifications may be considered as a hypothesis of load sharing, whose validity should be verified later when experimental data or results of a deformation analysis become available.

4.2 Normal Forces on Gear Teeth

The coordinates (x_{mp}, z_{mp}) in S_j and the parameter ψ_{mp} of the worm center point for the mth pair of meshing teeth can be found by solving Eqs. 35 and the following set of equations describing the gear pitch circle:

$$(a-x_j)^2 + z_j^2 = (d_2/2)^2$$

$$y_j = 0$$

(42)

The normal direction at the center point can be calculated from Eqs. 41, where ψ_{mp} is to be used for ψ, and θ^* is the given position.

24

From Fig. 4.1, where friction is not shown, the torque contributed by the normal force on the gear tooth of the mth meshing pair is:

$$T_{G_m} = F_{G_m}^n \, r_{e_m}$$

$$r_{e_m} = [\, (a - x_p) \frac{|n_{jz}|}{n_W} - z_p \frac{|n_{jx}|}{n_W} \,]_m \tag{43}$$

$$n_W = \sqrt{n_{jx}^2 + n_{jy}^2 + n_{jz}^2}$$

where r_{em} is an effective radius; n_{jx}, n_{jy}, n_{jz} are as given in Eqs. 41; $[\ ]_m$ denotes the mth meshing pair.

By hypothesis, F_{Gm}^n is proportional to the length of contact lines. Besides, $T_G = \Sigma T_{Gk}$. Hence the formula:

$$F_{Gm}^n = \frac{T_G \, L_m}{\displaystyle\sum_{k=1}^{N_p'} L_k \, r_{ek}} \tag{44}$$

where

F_{Gm}^n is the normal force acting on the gear tooth of the mth meshing pair,

T_G is the required output torque from the gear shaft,

L_m is the length of contact lines on the mth pair of the meshing teeth, and

N_p' is the actual number of meshing pairs.

The relative motion between the worm and gear is pure sliding. Hence, friction plays an important role in the performance of the wormgear (Shigley and Mitchell, 1983). Let μ be the Coulomb friction coefficient, then the magnitude of friction is μF_{Gm}^n, and its direction on the gear is the same as the direction of the relative velocity $v_1 - v_2$.

At the center point of the mth pair the worm velocity is parallel to the

(x_j, y_j) plane and equal to $x_{mp}\omega_1$, the gear velocity is in the (x_j, z_j) plane and equal to $d_2\omega_2/2$. In S_j the direction of the relative velocity is given by Eqs. 39 and is shown in Fig. 4.2.

Let $F_{fGm}{}^n$ be the normal force on the gear tooth of the mth meshing pair when friction is present. The torgue due to the friction force, $\mu F_{fGm}{}^n$, is:

$$T_{fm} = - F_{Gm}^n \, r_{efm}$$

$$r_{efm} = [\; \frac{\mu \dfrac{\omega_2 d_2}{2}}{\sqrt{(x_p\omega_1)^2 + (d_2\omega_2/2)^2}} \; \frac{d_2}{2} \;]_m \tag{45}$$

Then, $F_{fGm}{}^n$ is found to be:

$$F_{fGm}^n = \frac{T_G \, L_m}{\displaystyle\sum_{k=1}^{N_p} L_k \; (\, r_{ek} - r_{efk} \,)} \tag{46}$$

4.3 Tangential Forces on the Worm

The force acting on the worm can be decomposed into the radial, tangential, and axial components, viz. $F_{Wm}{}^{xj}$, $F_{Wm}{}^{yj}$, and $F_{Wm}{}^{zj}$ (Shigley and Mitchell, 1983). See Fig. 4.1. The tangential component balances the input worm torque, which in turn determines the input power. The ratio of the input power without friction to that with friction, averaged over an interval where the gear rotates 3.273 deg, determines the efficiency of the wormgear.

The tangential force on the worm for the mth pair of the meshing teeth is:

without friction

$$F_{Wm}^{yj} = - F_{Gm}^n \; (\; \frac{|p_{jy}|}{n_W} \;)_m \tag{47}$$

26

with friction

$$F_{fWm}^{yj} = F_{fGm}^{n} \left[\frac{\mu x_p \omega_1}{\sqrt{(x_p \omega_1)^2 + (d_2 \omega_2 /2)^2}} + \frac{|n_{jy}|}{n_W} \right]_m \qquad (48)$$

Since the worm torque equals the sum of the torques contributed from the
meshing teeth, it is:

without friction

$$T_W = \sum_{m=1}^{N_p'} F_{Wm}^{yj} x_{mp} \qquad (49)$$

with friction

$$T_{fW} = \sum_{m=1}^{N_p'} F_{fWm}^{yj} x_{mp} \qquad (50)$$

4.4 Results and Discussion

Load sharing was computed for a power output of 4000 kW and a rotation
speed of 130 r/min of the gear shaft. To account for the effect of friction
at the worm and gear contact, a Coulomb friction coefficient value of 0.03 was
used in the computation. The results are shown in Figs. 4.3 to 4.5. In these
figures, the stars represent the case with friction and the circles the case
without friction.

Figs. 4.3(a) to 4.3(c) present the normal forces acting on the meshing
gear teeth for the three wormgear positions, viz. $\theta^* = 0$, 1 and 2 deg. These
figures correspond to the cases of Figs. 3.8(a) to 3.8(c). The variation of
the normal force on one gear tooth as it moves through the meshing zone is
shown in Fig. 4.4, which is essentially the three graphs of Figs. 4.3
combined. It is seen that the difference between including and not including
friction cannot be detected. This implies that, for a given gear output
power, the lubricant film at the worm and gear contact must withstand almost

27

the same normal force whether friction is present or not.

The tangential forces acting on the worm thread corresponding to the three wormgear positions of $\theta^* = 0$, 1 and 2 deg are shown in Figs. 4.5(a) to 4.5(c). Here, including or not including friction makes a big difference. This is because for the selected wormgear the direction of the relative velocity, which is the direction of friction, is very close to that of the tangential force on the worm. Thus, a large portion of the input torque must be spent to overcome the frictional torque. Hence, effective lubrication at the contact can drastically reduce the input power required of the worm.

5 Description of Computer Programs

Two computer programs were developed in this work, the graphics program and the analyses program. The graphics program is for creating the 3-D graphics of the selected wormgear. The analyses program is for implementing the geometric, kinematic and load sharing analyses.

5.1 Graphics Program

The graphics program is named NASA.PRG and is written in I-DEAS programming language, IDEAL.

5.1.1 I-DEAS Software

I-DEAS (Integrated Design Engineering Analysis System) is a widely used commercial software (SDRC, 1991). It has three program families: solid modeling, finite element and advanced technique. Solid modeling is used to geometrically create an object. The tasks in the solid modeling family include construction geometry, object modeling, mechanism design, drawing

28

layout, etc.

Construction geometry is used to create profiles, wireframes and skin groups. A profile is defined by a boundary that lies in a plane. The boundary can consist of lines, arcs, circles and splines, which can be defined by points, centers and radii. In the boundary holes can be defined.

Object modeling is used to design mechanical parts which are modeled as objects. There are several ways to create an object: (1) Use primitives such as blocks, cylinders, cones and spheres. (2) Extrude or revolve a profile into a solid. (3) Create a group of profiles and then generate a solid through the group. This task also provides jointing, cutting and assembling options. Once two objects are created, a third object can be created from them by using these options. The created object can be displayed with options such as scale, hidden line, shade display, etc. The created object can also be rotated, moved and oriented to the required position; and saved and printed.

I-DEAS can be run interactively by selecting from the menu on the screen using the mouse, or by inputing from the keyboard or through a program.

I-DEAS has a high level programming capability using the language IDEAL. The commands can be written in a program file which is then executed. The programming capability permits arithmetic expressions, variables, mathematic functions, and algebraic operations; it also provides commands for branching and do-looping, as well as controlling the input and output.

5.1.2 Description of the Graphics Program

The process of creating a double-enveloping wormgear by using I-DEAS is similar to that of manufacturing the wormgear. This is carried out as

29

follows:

First a worm blank and a disc grinding wheel are created. Once these two objects are created, the worm can be generated by cutting the worm blank with the grinding wheel. In the worm creating process the motion of the worm blank and the grinding wheel is similar to that of the first enveloping process.

Then the gear blank is created. The gear is created by cutting the gear blank with the previously created worm. The motion of the gear blank and the worm is similar to that of the second enveloping process.

The motions are simulated by rotating the blanks and cutters about their respective axes. The rotation is divided into incremental steps. At each step the cutting operation takes place. The incremental step determines how smooth the generated worm and gear surfaces are. As a result there is a difference between the I-DEAS generated surface and the manufactured surface. Because of the computer storage limitation, the steps can not be chosen too small.

The structure of the graphics program is shown by the flow chart in Fig. 5.1, which consists of the following steps:
1. Input wormgear dimensions and parameters which are needed in its manufacture.
2. Input the incremental stepsizes ('dtw' for generating the worm and 'dtg' for generating the gear) and the total number of steps in the cutting processes ('wn' for generating the worm and 'gn' for generating the gear).
3. Construct the grinding wheels for generating the worm. Four are constructed and jointed in sequence to reduce the CPU time, i.e. four coils of the worm thread are created for one revolution of the worm

blank (dtw*wn=360 deg).

4. Create the worm and gear blanks.

5. Create the worm. When the process is completed, the created worm is saved under the name 'worm'.

6. Create the gear. When the process is completed, the created gear is saved under the name 'gear'.

7. Display the created worm and gear.

The program is listed in Appendix 1. The program works interactively and ends at the object modeling screen. This allows the user to see the created objects again before exiting I-DEAS.

The procedure to execute the program is as follows: (1) Enter I-DEAS and come to the construction geometry task screen. (2) Run program in the construction geometry screen. More detailed instructions are given in Appendix 1.

Because the worm thread of the double-enveloping wormgear is asymmetric about its center, the number of steps needed to create the whole worm is larger than that for creating other machine parts, such as the spur gear or the cylindrical worm. More coils of the worm thread requires more steps. Every step increases the number of surfaces on the worm blank. The larger the number of surfaces, the more CPU time and computer disc space are needed to calculate and save the data. Once the capacity of the computer or the I-DEAS space is full, the screen stops, or the program crashes. This is why there are only four coils of worm thread shown in Fig. 2.7. The program was executed on an IBM 5080 terminal at SUNY-Binghamton.

From the above description it can be seen that (1) the computer simulation is helpful to visualize the selected wormgear, and (2) the computer simulation can not replace the analysis, because the numerical data of the

generated surfaces are not accurate enough for the projected lubrication analysis.

5.2 Analyses Program

5.2.1 Overall view of the Program

The analyses program is named NASA.FOR and is written in FORTRAN. This program consists of a main program and several subroutines. The main program is the driver program of all the subroutines and may be tailored and adapted to the final design package. The subroutines fall in two categories: those driven by the main program (DIMEN, WORM, WORML, GEAR, CONTACT, PITCH and LOAD) and those driven by other subroutines (EQW, EQWL, EQG, FIRST, SECOND, WORMT, ROOT and NORMAL). Most of the logical judgements are done in the subroutines called by the main program. The evaluations of analytical expressions are done in the subroutines called by other subroutines.

The structure of the analyses program is shown by the flow chart in Fig. 5.2, which consists of the following steps:
1. Calculate the wormgear dimensions (DIMEN).
2. Calculate the coordinates in S_1 of the worm surface and the curvatures of the worm surface (WORM and WORML).
3. Calculate the coordinates in S_2 of the points on the gear surface along the contact line and the curvatures of the gear surface (GEAR).
4. Calculate the contact lines in S_n on every pair of the meshing teeth for a given position (CONTACT).
5. Calculate the coordinates in S_j and the kinematic properties of the worm center points for a given position (PITCH).
6. Calculate the load distribution on the meshing teeth for a given position (LOAD).

Input of this program, under the 'parameter' command, are at the beginning of the main program and are explained there. In the main program the given position θ^* is named 'tn2', which should be the input for the subroutines CONTACT, PITCH and LOAD.

Input to a subroutine are explained at the beginning of the subroutine. The output to be used by other subroutines are described in the subroutine statements. The output not to be used by other subroutines are stored in data files (DIMEN, WORM, WORML, GEAR, CONTC, PITCH and LOAD).

Error flags would be shown on the screen such as 'dimension is not big enough', 'the stepsize is too large', 'the worm surface intersects the edge curve', etc.

5.2.2 Detailed Description of the Program

Main Program

The input under the 'parameters' command include the center distances a_0 and a, the speed ratios i_0 and i, the number of worm thread N_1, the number of meshing teeth N_p, the worm pitch diameter d_1, the apex angle β, the rotation speed of the worm, the output power of the wormgear, and the friction coefficient at the contact.

The stepsizes and the dimension arrays are input under the 'data' command. The input stepsizes are: 'df' (for the running parameter ψ_1), 'dt' (for the running parameter θ_2), 'dx' (in the x axis), and 'dz' (in the z axis). There are four one-dimensional common arrays with dimension nn=600.

The given position θ^* is named 'tn2'. The wormgear position of $\theta^*=0$ is shown in Fig. 3.6. 'tn2' is not to be larger than the angle between two

33

neighboring gear teeth. The two angles 't20' and 't21' specify the interval of the running parameter θ_2, which is chosen to be slightly larger than the meshing zone.

The convergence criteria for root search are specified by 'ep' (for angles) and 'epp' (for distances). A logical judgement for the number of meshing teeth is set up in the main program. This is because the actual number of meshing teeth obtained from subroutines CONTACT and PITCH should agree, otherwise the results can not be used in the subroutine LOAD. A flow chart for the main program is shown in Fig. 5.3.

Subroutine DIMEN

This subroutine must be run before any other subroutine, because it provides the wormgear parameters and dimensions. The formulas used for the calculation can be found in AGMA (1965), Dudley (1984), and Shigley and Mitchell (1983).

The meshing zone is specified by 'start' and 'end', which are the angles where a gear tooth runs into and out of the meshing zone respectively. 'f20' and 'f21' are the two angles specifying the interval of the running parameter ψ_2. 'tao' is the angle between two neighboring gear teeth. The half gear face angle, named 'dlt', is used to specify the ψ_1 interval in the calculation of the gear surface. 'reps' is an angle for determining the position of the original generating plane that grinds the other side of the worm thread, Fig. 5.4.

Subroutine WORM and WORML

These subroutines implement the analysis described in Sections 3.3 and 3.5. The two sides of the worm thread are calculated by WORM and WORML,

34

respectively.

Three logical judgements are set up. The first checks if the worm surface intersects the edge curve. The second checks which branch of the envelope surface is the real worm surface, Fig. 3.2. The third examines the tooth domain.

Subroutines EQW and EQWL are called by subroutines WORM and WORML, respectively. A flow chart for WORM (and WORML) is shown in Fig. 5.5.

Subroutine GEAR

This subroutine implements the analysis described in Sections 3.4 and 3.5. Calculations for both the conventional form and the modified form are provided.

A θ value, named 'tl', is first given. Then ψ, named 'fl', is varied near the θ value to calculate the coordinates of the points on the gear surface. This is because if ψ is far away from θ, it is also off the tooth domain. The range of ψ variation is twice the gear face angle. This range is much less than the range of the running parameter ψ in the subroutine WORM. Hence, the stepsize $\Delta\psi_1$, 'df', used here is a tenth of the stepsize given in the main program.

The logical judgements set up in this subroutine consist of the tooth domain examinations and sorting out the fictitious points. A flow chart for GEAR is shown in Fig. 5.6.

Subroutine CONTACT

This subroutine implements the analysis of Section 3.6.1. Calculations

for both the conventional form and the modified form are provided. At a given position 'tn2', the contact lines on all the meshing teeth are calculated. The angles 'start' and 'end' are used in this subroutine to find out how many pairs of meshing teeth actually exist in the meshing zone.

Similar to the case of GEAR, ψ is varied near the θ value. To obtain accurately the length of the contact lines, as many points are needed as is practicable. Hence the stepsizes of ψ and z are chosen to be df/100 and dz/10, where 'df' and 'dz' are the input stepsizes from the main program.

Four temporary one-dimensional arrays are used to save the coordinates of the points on the same contact line. 'nn' gives the dimension of these arrays, which is controlled by the main program. If the saved points exceed 'nn', an error flag would appear on the computer screen. The number 'nc' (output name 'ncc') records how many pairs of meshing teeth exist in the meshing zone for the given position, which should be the same as that obtained from the subroutine PITCH.

Output of the contact line length is stored in the array 'CONTC', which has dimension N_p and is the input of the subroutine LOAD. A flow chart for CONTACT is shown in Fig. 5.7.

Subroutine PITCH

This subroutine implements the analysis of Sections 3.4 and 3.6.2. At the given position 'tn2' (named 'fn2' here), the worm center point on each pair of the meshing teeth are obtained by first finding the points on the worm profile in the mid-plane, i.e. $y_j = 0$, then sorting out the center point on the profile by using the subroutine ROOT. The angles 'start' and 'end' are used to check if a worm coil falls in the meshing zone.

36

To obtain accurately the center point coordinates while still saving the CPU time, the stepsize is chosen to be df/100, where 'df' is the input stepsize from the main program. The range of ψ variation is 90 deg around the given position 'fn2'. The index mw=1 would mean that the worm profile intersects the edge curve. The coordinates of the center points obtained are in S_j. These coordinates and the normal directions at the center points are stored in one-dimensional arrays with the maximum dimension N_p, and are input for the subroutine LOAD.

The number 'nc' (output name 'ncp') records how many pairs of meshing teeth exist in the meshing zone for the given position, which should be the same as that obtained from the subroutine CONTACT. A flow chart for PITCH is shown in Fig. 5.8.

Subroutine LOAD

This subroutine implements the analysis of Sections 4. Before this subroutine is run, the subroutines PITCH and CONTACT must be run so as to provide the input for this subroutine.

Subroutine ROOT

This subroutine searchs the root in a given interval where only one root exists. The root searching is based on the idea of bisection (Press, 1986), but three sections are used instead to reduce the CPU time. Here the root is a point at which the radius of the worm is equal to the radius of the worm pitch cylinder at the same axial position. The criteria used to search the center point are the difference of the two radii or the angles between the two grid points.

37

6 Summary

Based on the consideration of achieving better contact properties (longer contact lines and a larger angle between the contact line and the relative sliding velocity), a double-enveloping wormgear was selected. The worm was generated by a disk grinding wheel. The center distance and the speed ratio were respectively the same in the first and the second enveloping processes (conventional form).

Based on the suggested speed ratio and center distance for helicopter applications, the other dimensions and geometric parameters of the wormgear were laid out basically in accordance with the AGMA gear standards. However, some of these quantities were modified as a result of compromising between conflicting requirements. It is felt that for a novel application like ours that is not covered by past experiences, some deviations from the common practice would be inevitable.

3-D graphics representation of the selected wormgear was developed using the I-DEAS software. The software works like a cutting tool generating a surface from a blank, but in discrete steps. Because of the computer storage limitation, the steps could not be made too small. As a result, the numerical data of the generated surfaces were not accurate enough for the projected lubrication analysis. However, the graphics representation does convey a clear notion of the proportion of the selected wormgear.

The kinematics of meshing was analyzed with the method of enveloping. The method is simple and elegant, and can be used to construct conjugate surfaces in other contacting mechanisms as well. Using this method the shape of the worm and gear surfaces, the contact lines, the normal directions, the relative normal curvatures, and the relative sliding velocities at the contact points were obtained. With these results the clearance function between the

38

meshing surfaces can also be obtained, which is needed for the projected lubrication analysis.

Based on the contact line pattern revealed by the kinematics analysis, a hypothesis of load sharing was made. Using this hypothesis the variation of the normal contact force acting on a gear tooth as it ran through the meshing zone, as well as the distribution of the normal forces on all the meshing gear teeth at any instant, were obtained. This information is needed in the projected lubrication analysis. The required torque output of the worm was likewise obtained. The results clearly demonstrated the advantage and need of implementing effective lubrication at the contact.

A computer program, written in FORTRAN, was developed to carry out the computations. The structure of the computer program and its usage are described in detail.

References

American Gear Manufactures Association, 1965, "Design of General Industrial Double-Enveloping Wormgears", AGMA Standard 342.02.

Chaiko, L., 1990, "Assessment of Worm Gearing for Helicopter Transmissions", NASA Technical Memorandum 102441, AVSCOM Technical Memorandum 89-C-010.

Dudley, D. W., 1984, Handbook of Practical Gear Design, McGraw-Hill.

Elkholy, A. H., 1989, "Load Distribution on Contact Lines of Helical Gear Teeth", Proceedings of the 1989 International Power Transmission and Gearing Conference, pp. 135-142.

Huston, R. L. et al., 1989, "Computer Aided Design of Spur Gear Teeth", Proceedings of the 1989 International Power Transmission and Gearing Conference, pp. 539-545.

Kreyszig, E., 1959, Differential Geometry, University of Toronto Press.

Litvin, F. L., 1989, Theory of Gearing, NASA Reference Publication 1212.

Press, W. H. et al., 1986, Numerical Recipes, Cambridge University Press.

SDRC (Structural Dynamics Research Corporation), 1991, I-DEAS Solid Modeling User's Guide.

Shen, Y. F. et al., 1983, Meshing of Spatial Mechanisms and the SG-71 Wormgear (in Chinese), Metallurgy Industry Publication Corporation, Beijing.

Shigley, J. E. and Mitchell, L. D., 1983, Mechanical Engineering Design, 4th Ed., McGraw-Hill.

Simon, V., 1989, "A New Type of Ground Double Enveloping Worm Gear Drive", Proceedings of the 1989 International Power Transmission and Gearing Conference, pp. 281-288.

Steward, J. H., 1989, "The Compliance of Solid, Wide-Faced Spur Gears", Proceedings of the 1989 International Power Transmission and Gearing Conference, pp. 181-188.

Table 1 Dimensions of the Selected Wormgear

TERM	SYMBOL	VALUE	FORMULA BASED	REFERENCE
speed ratio	i	110		given
normal pressure angle	α_n	20 deg		Dudley, 1984 p. 3.71
center distance	a	880 mm		selected
number of worm threads	N_1	1		selected
number of gear teeth	N_2	110	$N_2 = N_1*i$	
pitch diameter of worm*	d_1	300 mm	$d_1 = a^{0.875}/k_d$ (E) ($k_d = 1.7 \sim 2.2$)	Dudley, 1984 p. 3.70
pitch diameter of gear	d_2	1460 mm	$d_2 = 2*a-d_1$	
axial pitch	p_x	41.6975 mm	$p_x = \pi*d_2/N_2$	Dudley, 1984 Table 3.32
worm lead angle*	γ	2.533 deg	$\tan\gamma = p_x*N_1/\pi/d_1$	Dudley, 1984 Eq. 3.38
normal cirlular pitch	p_n	41.6568 mm	$p_n = p_x*\cos\gamma$	Dudley, 1984 Table 3.32
axial pressure angle*	α_x	20.018 deg	$\tan\alpha_x = \tan\alpha_n/\cos\gamma$	Dudley, 1984 Eq. 3.40
	$\Delta\alpha$	0.818 deg	$\Delta\alpha = p_x/2/d_2$	Dudley, 1984 p. 3.72
base circle diameter	d_b	519.3182 mm	$d_b = d_2*\sin(\alpha_x+\Delta\alpha)$	Dudley, 1984 Eq. 3.41
module	x_m	13.2727 mm/tooth	$x_m = p_x/\pi$	Dudley, 1984 Table 3.32
number of meshing teeth	N_p	10		selected
half angle of meshing	ψ_a	15.627 deg	$\psi_a = \pi*(N_p-0.45)/N_2$	Shen, 1983 Table 10.1
start angle of meshing	ψ_f	5.209 deg	$\psi_f = \sin^{-1}(d_b/d_2)-\psi_a$	Shen, 1983 Table 10.1
thickness of worm thread	S_1	18.7639 mm	$S_1 = 0.45*p_x$	Dudley, 1984 p. 3.73
thickness of gear tooth	S_2	22.9336 mm	$S_2 = 0.55*p_x$	Dudley, 1984 P. 3.73

whole depth	b_t	20.8284 mm	$b_t = 0.5*p_n$	Dudley, 1984 Table 3.31
working depth	b_k	18.7455 mm	$b_k = 0.45*p_n$	Dudley, 1984 Table 3.31
addendum	b_a	9.3728 mm	$b_a = 0.225*p_n$	Dudley, 1984 Table 3.31
clearance	c	2.0829 mm	$c = b_t-b_k$	
throat diameter of worm*	d_{a1}	318.7455 mm	$d_{a1} = d_1+2*b_a$	Shen, 1983 Table 10.1
root diameter of worm*	d_{f1}	277.0888 mm	$d_{f1} = d_{a1}-2*b_t$	Dudley, 1984 Eq. 3.39
throat diameter of gear	d_{a2}	1478.7455 mm	$d_{a2} = d_2+2*b_a$	Dudley, 1984 Table 3.32
root diameter of gear	d_{f2}	1437.0888 mm	$d_{f2} = d_{a2}-2*b_t$	Shen, 1983 Table 10.1
face width of worm	b_1	393 mm	$b_1 = d_2*\sin\psi_a$	Shen, 1983 Table 10.1
face width of gear	b_2	249 mm	$b_2 = (0.9\text{-}1.0)*d_{f1}$	Shen, 1983 Table 10.1
apex angle of generating plane	β	5 deg		selected

---

* At the center of worm.

Figures

43

44

Fig. 2.1 Single-enveloping wormgear (cylindrical wormgear)

Fig. 2.2 Double-enveloping wormgear

Fig. 2.3 Worm ground by a disc grinding wheel

Fig. 2.4 Worm ground by a cone grinding wheel

48
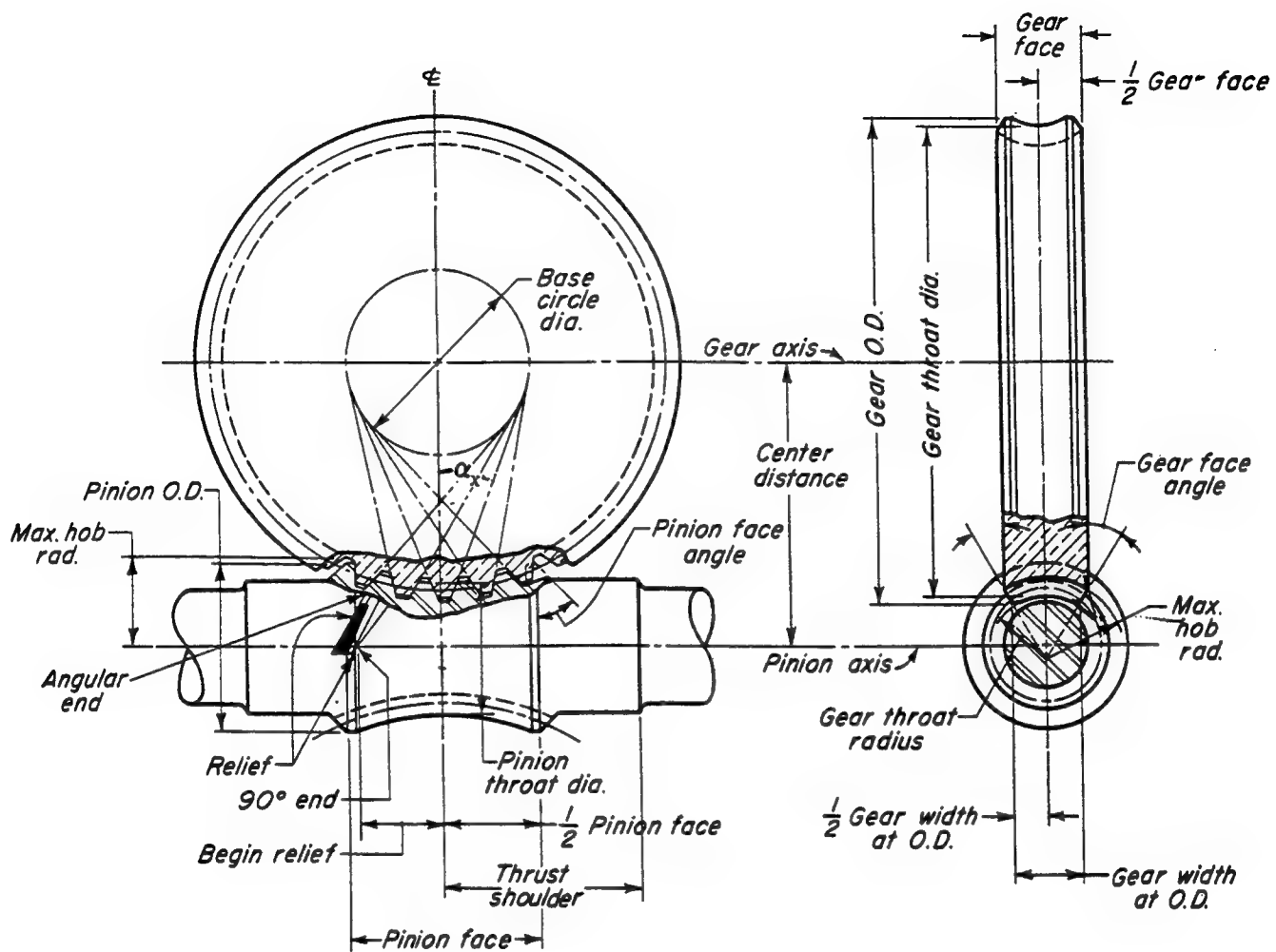
Fig. 2.5 Double-enveloping wormgear layout and nomenclature
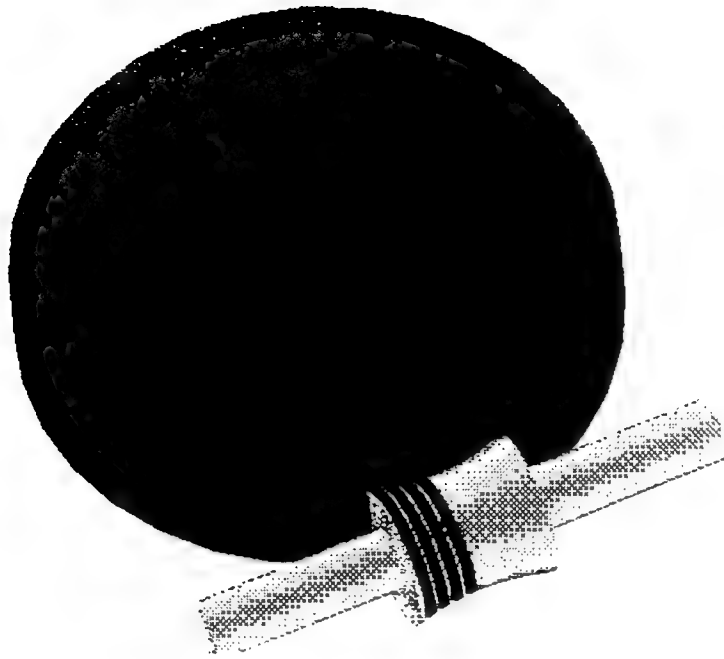
Fig. 2.6 Selected wormgear
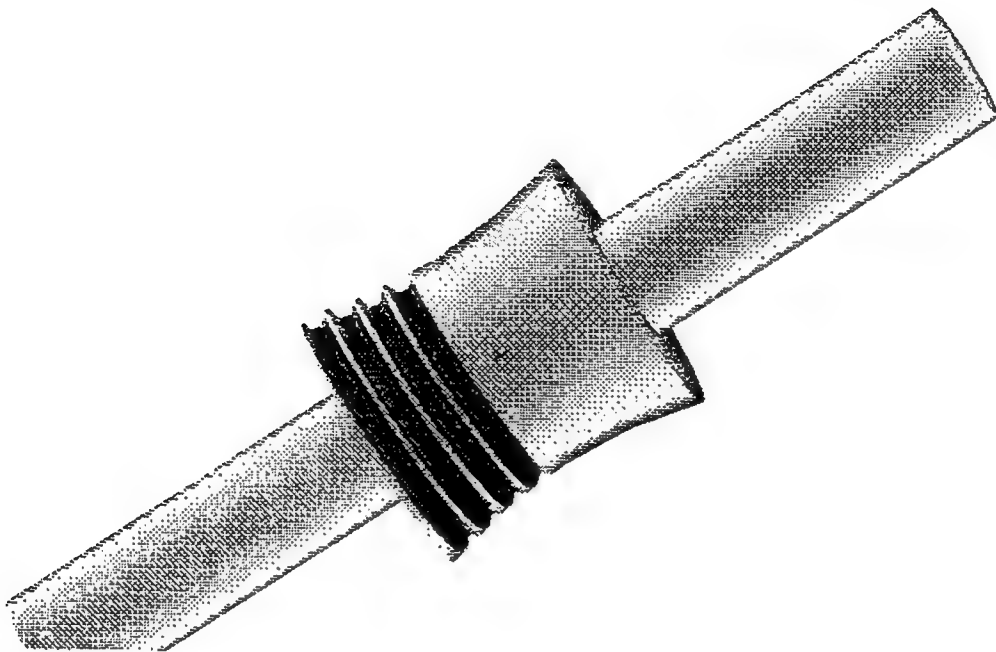
Fig. 2.7 Selected worm
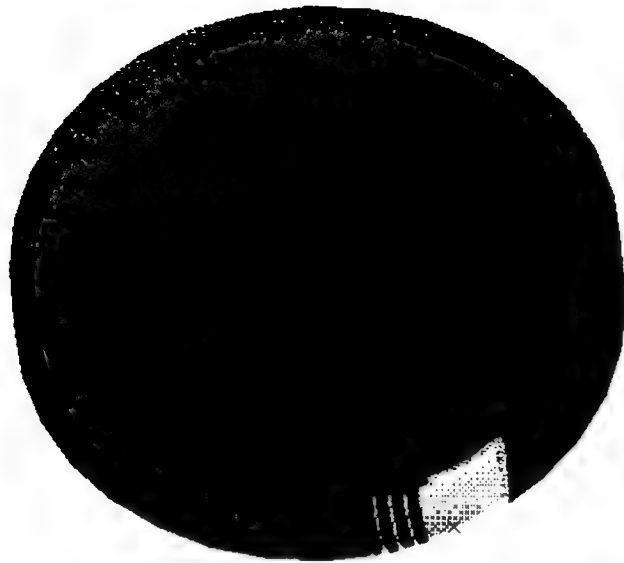
Fig. 2.8 Selected gear
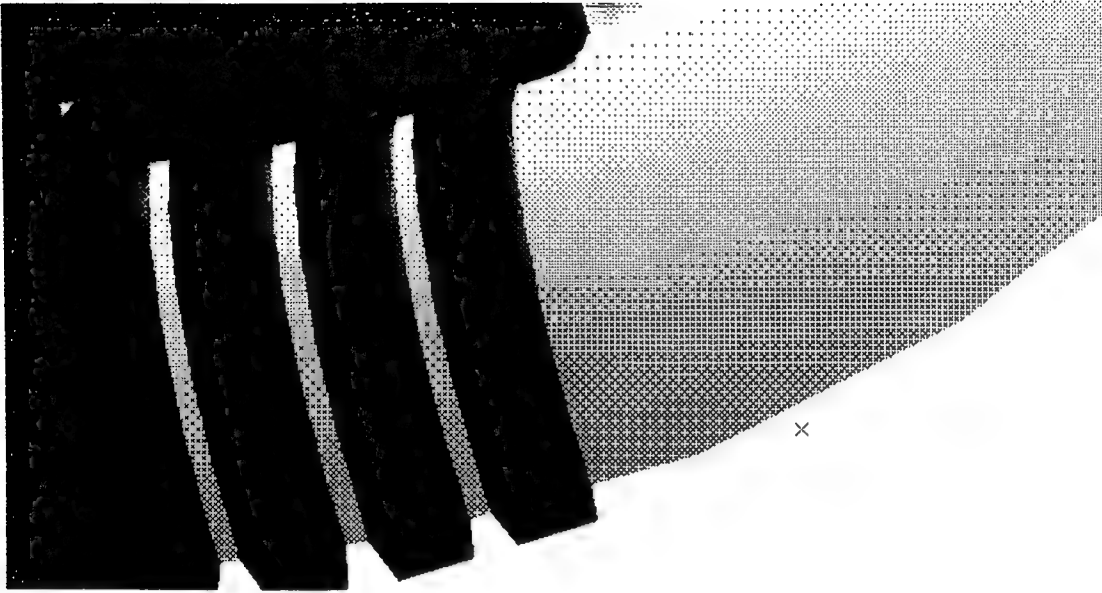
Fig. 2.9 Close-up view of the selected gear teeth
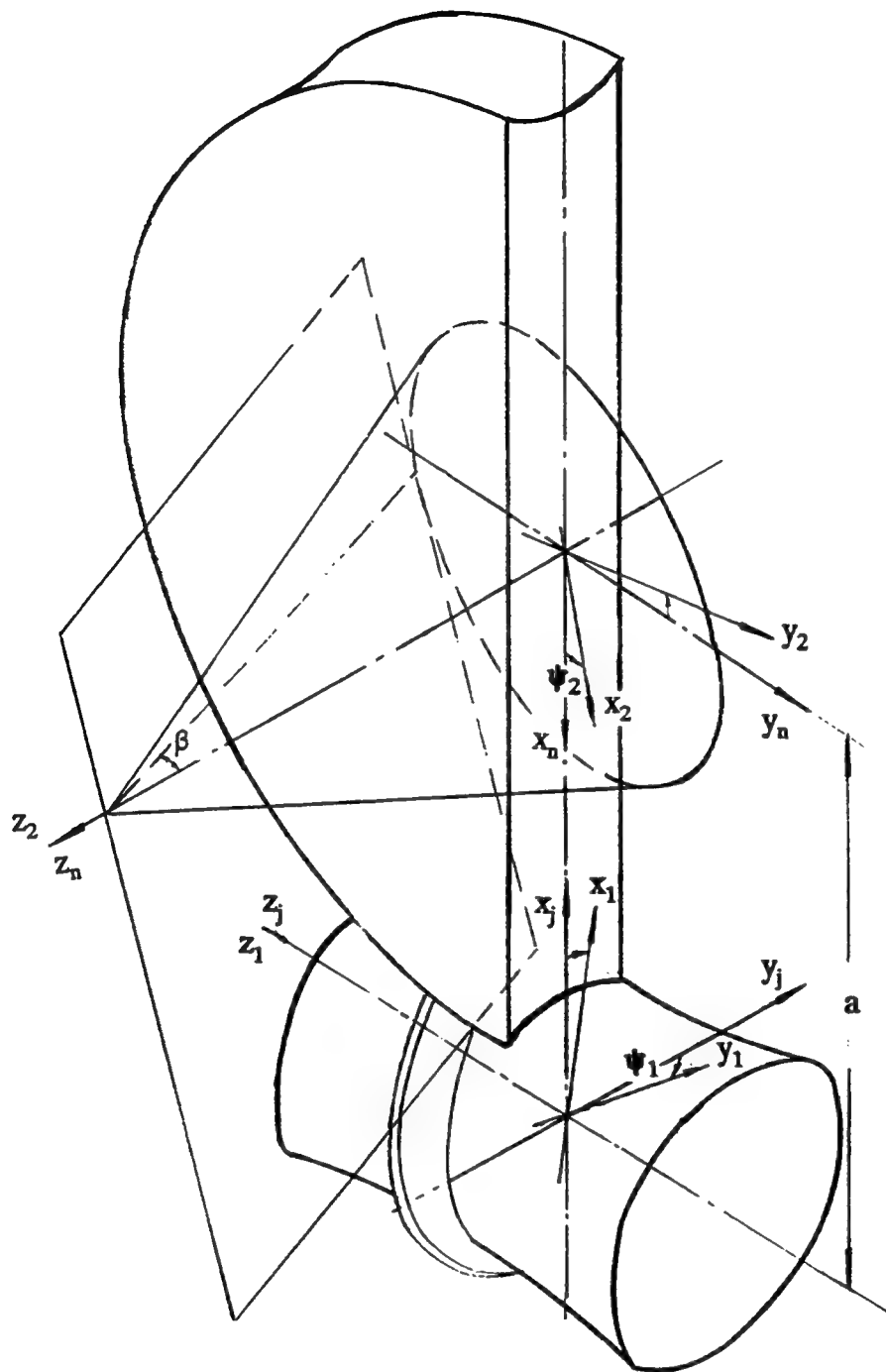
Fig. 3.1 Coordinate systems
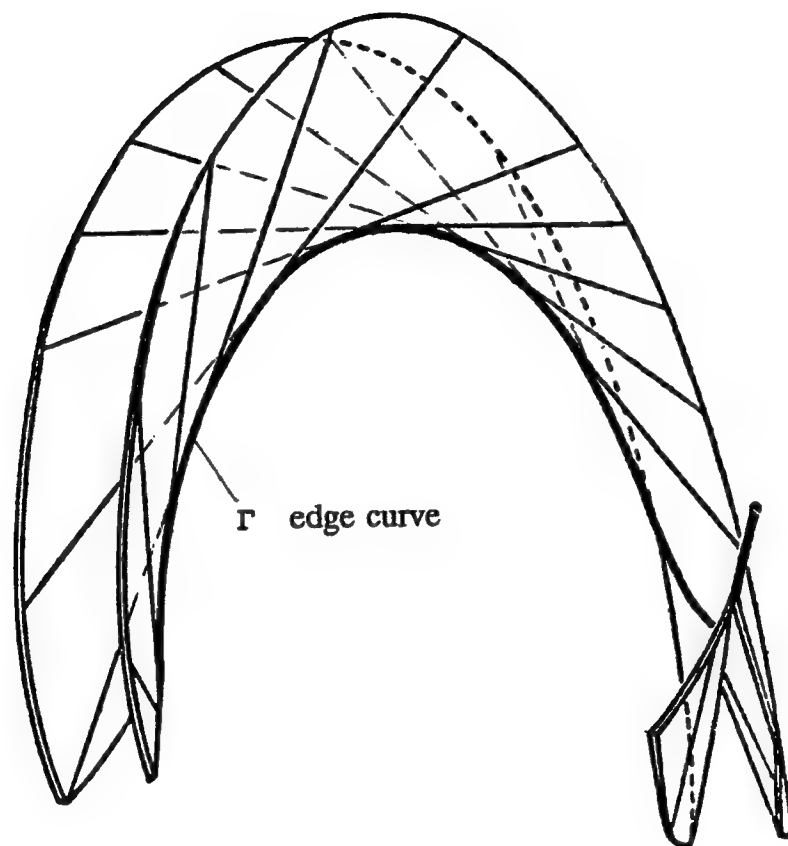
Γ   edge curve

Fig. 3.2 Envelope surface (worm) and its edge curve
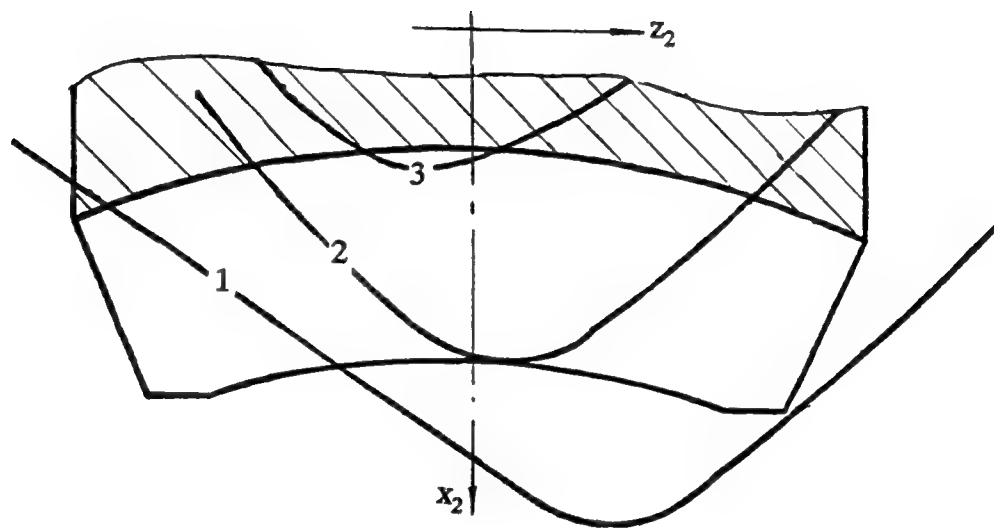
Fig. 3.3 Contact lines on a gear tooth of the modified form

56
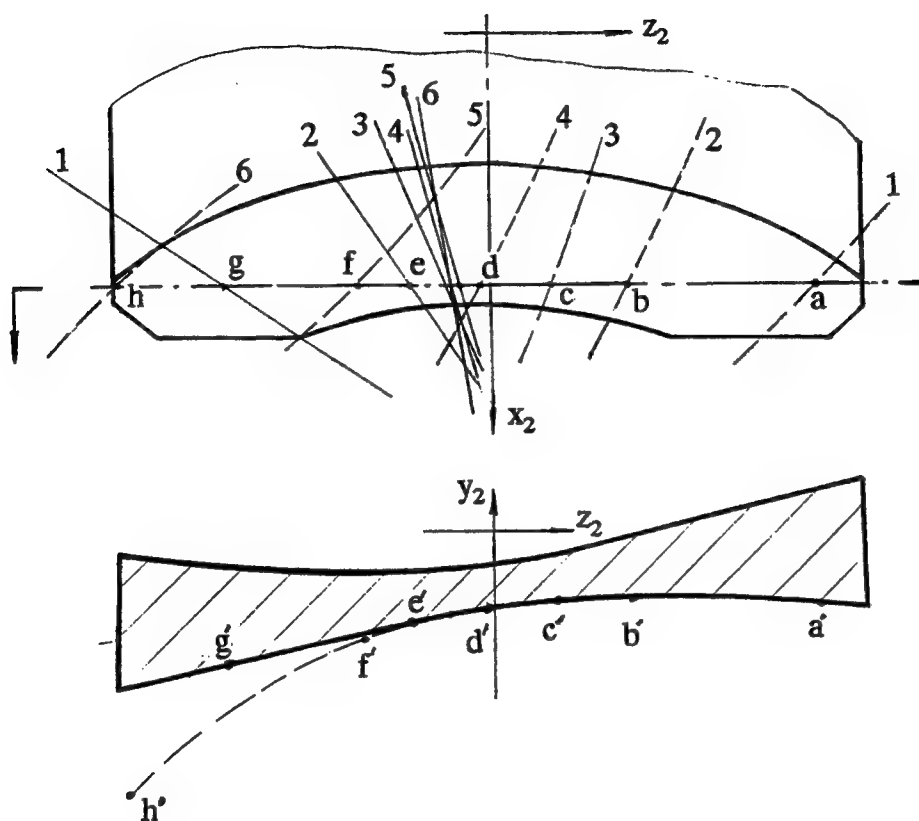
Fig. 3.4 Contact lines on a gear tooth of the conventional form

57
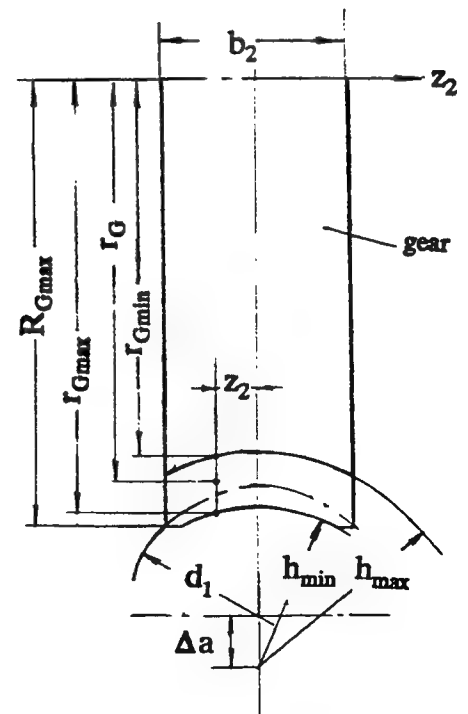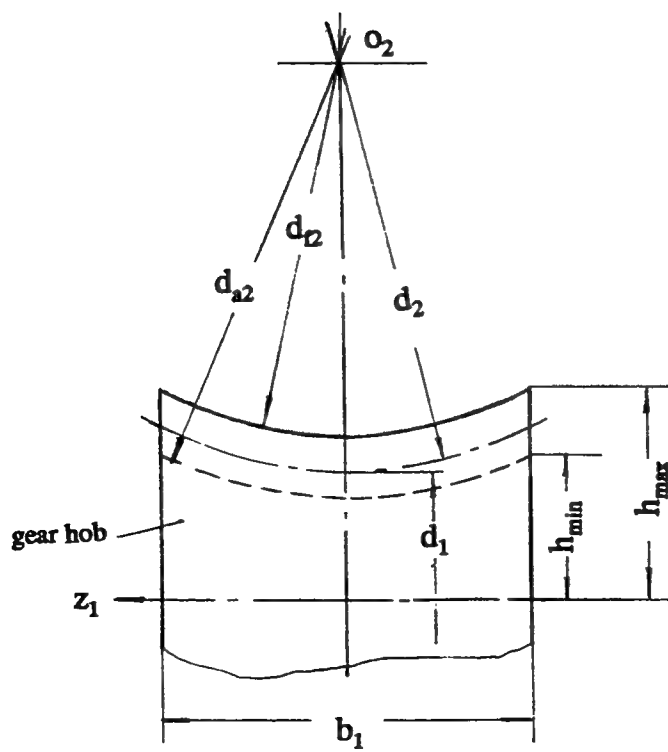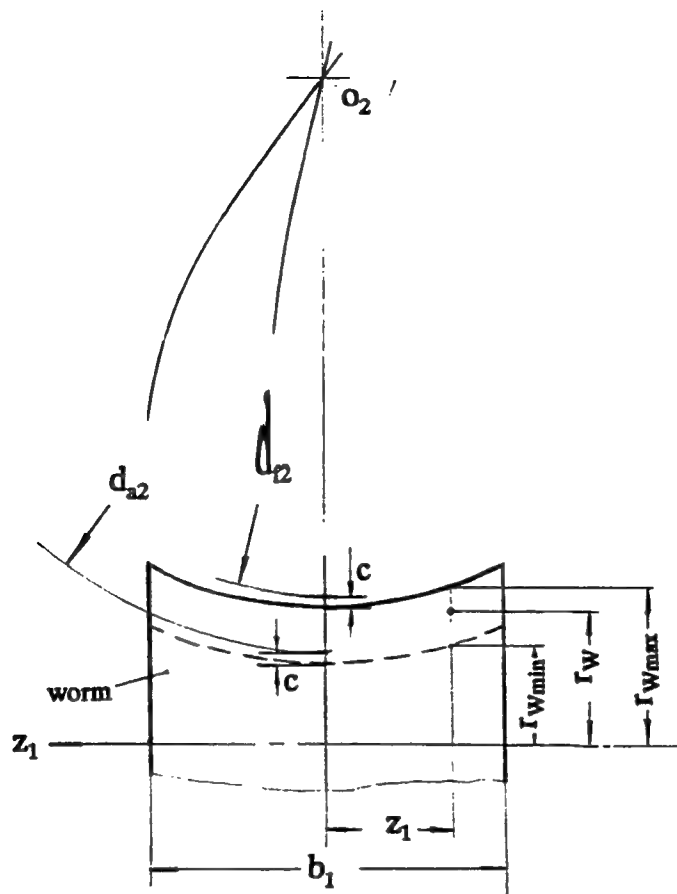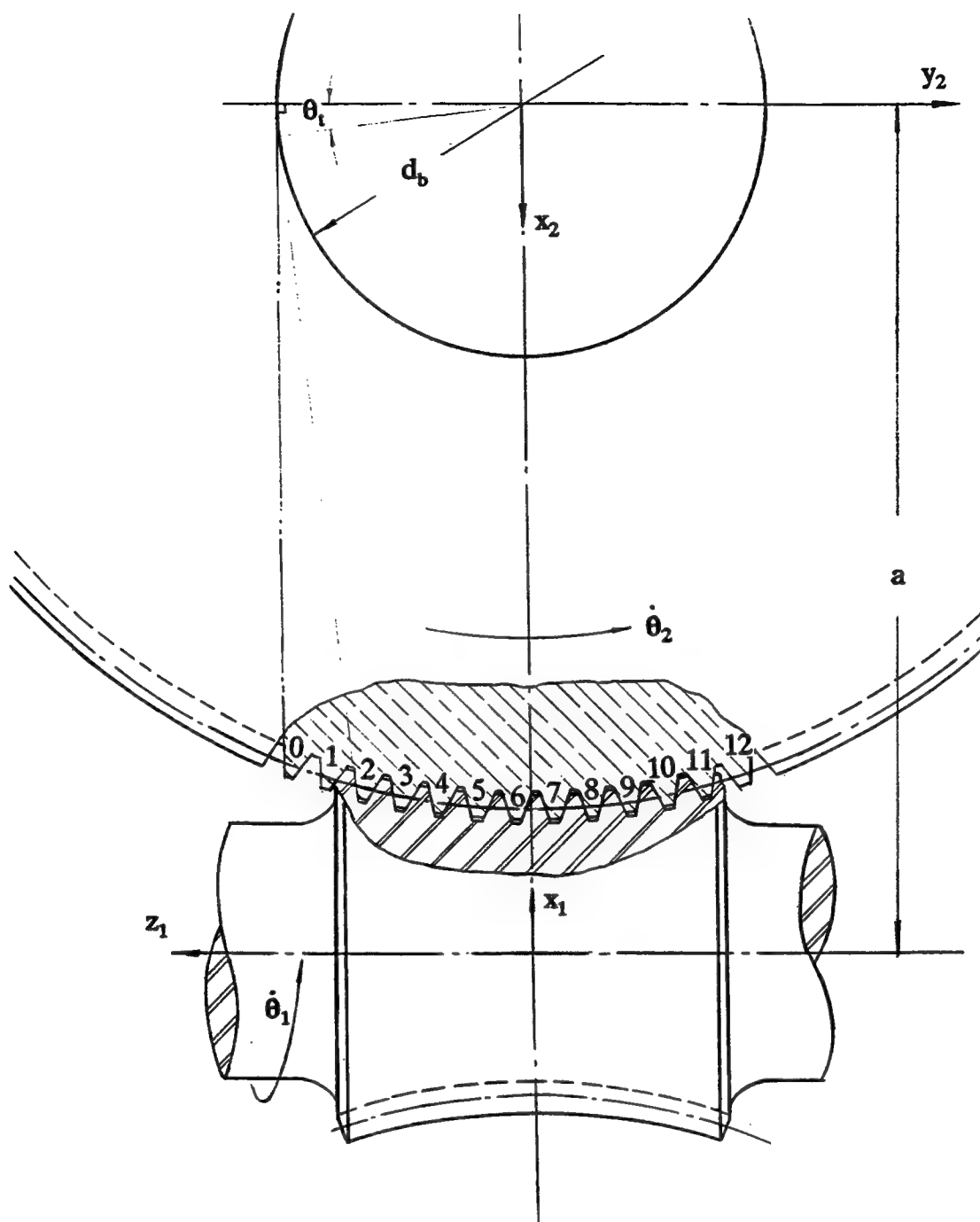
Fig. 3.6 Wormgear position at $\theta^* = 0$
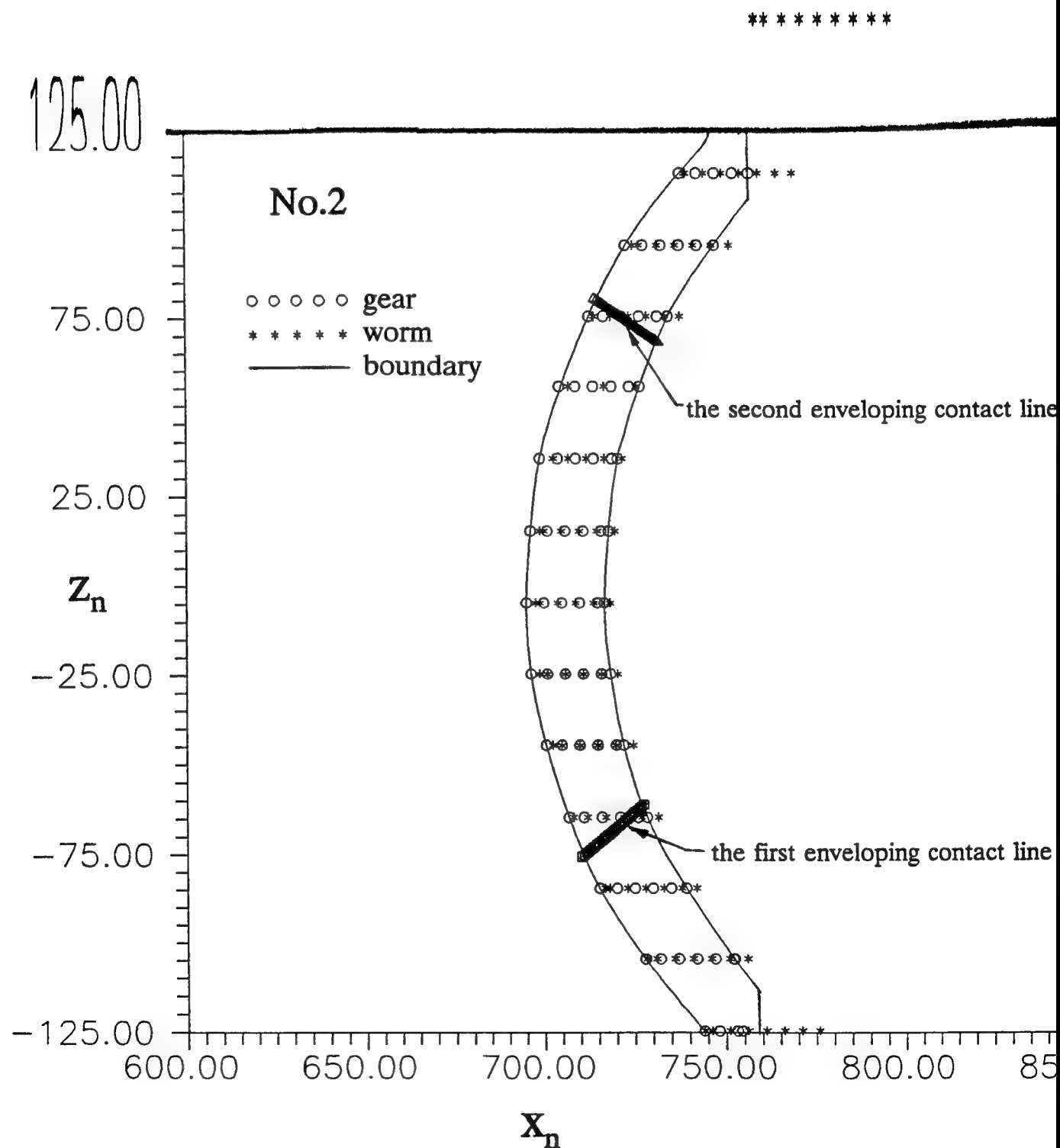
Fig. 3.7   Contact lines on the meshing gear teeth (a)
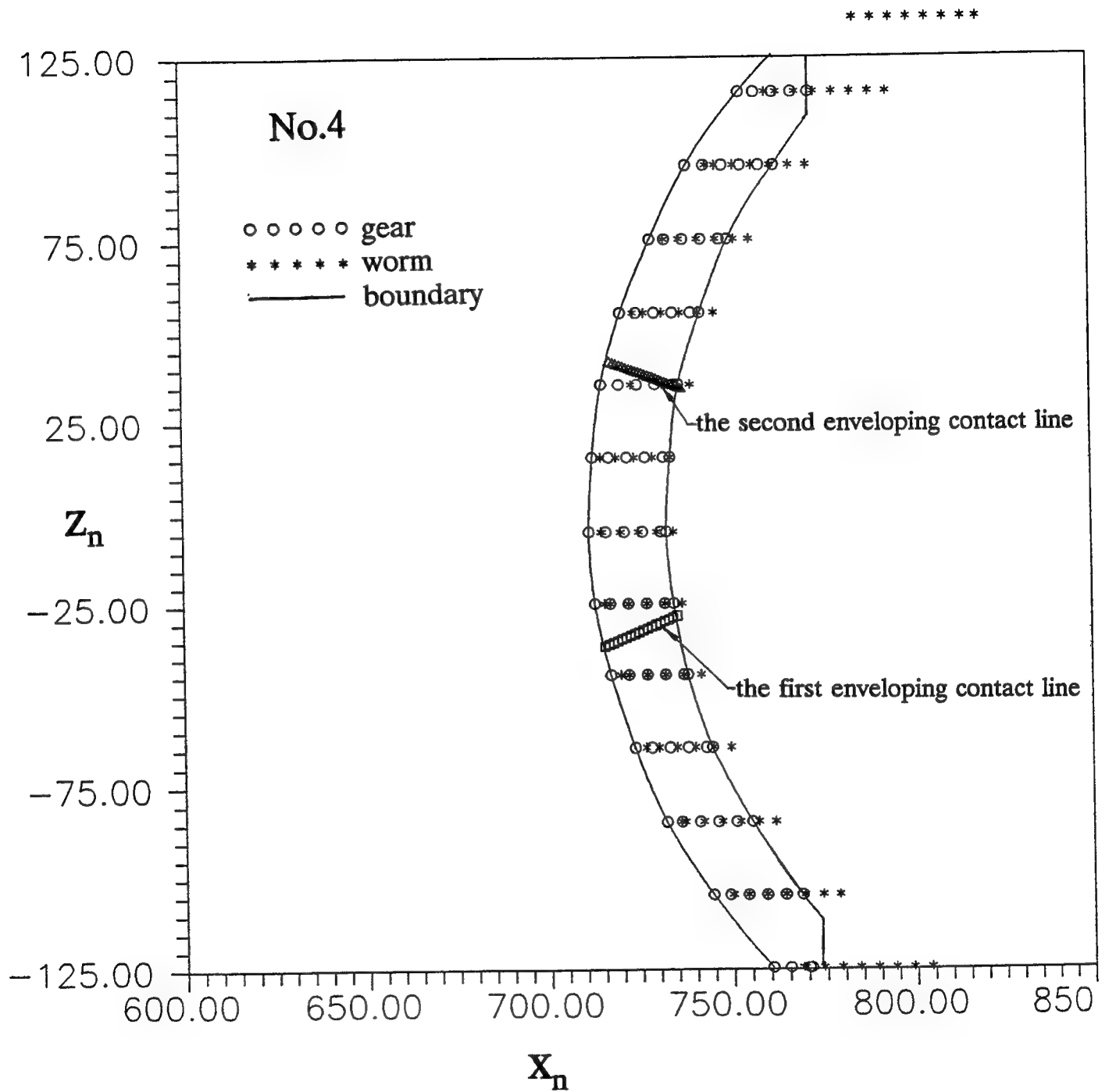
Fig. 3.7　Contact lines on the meshing gear teeth (b)

61
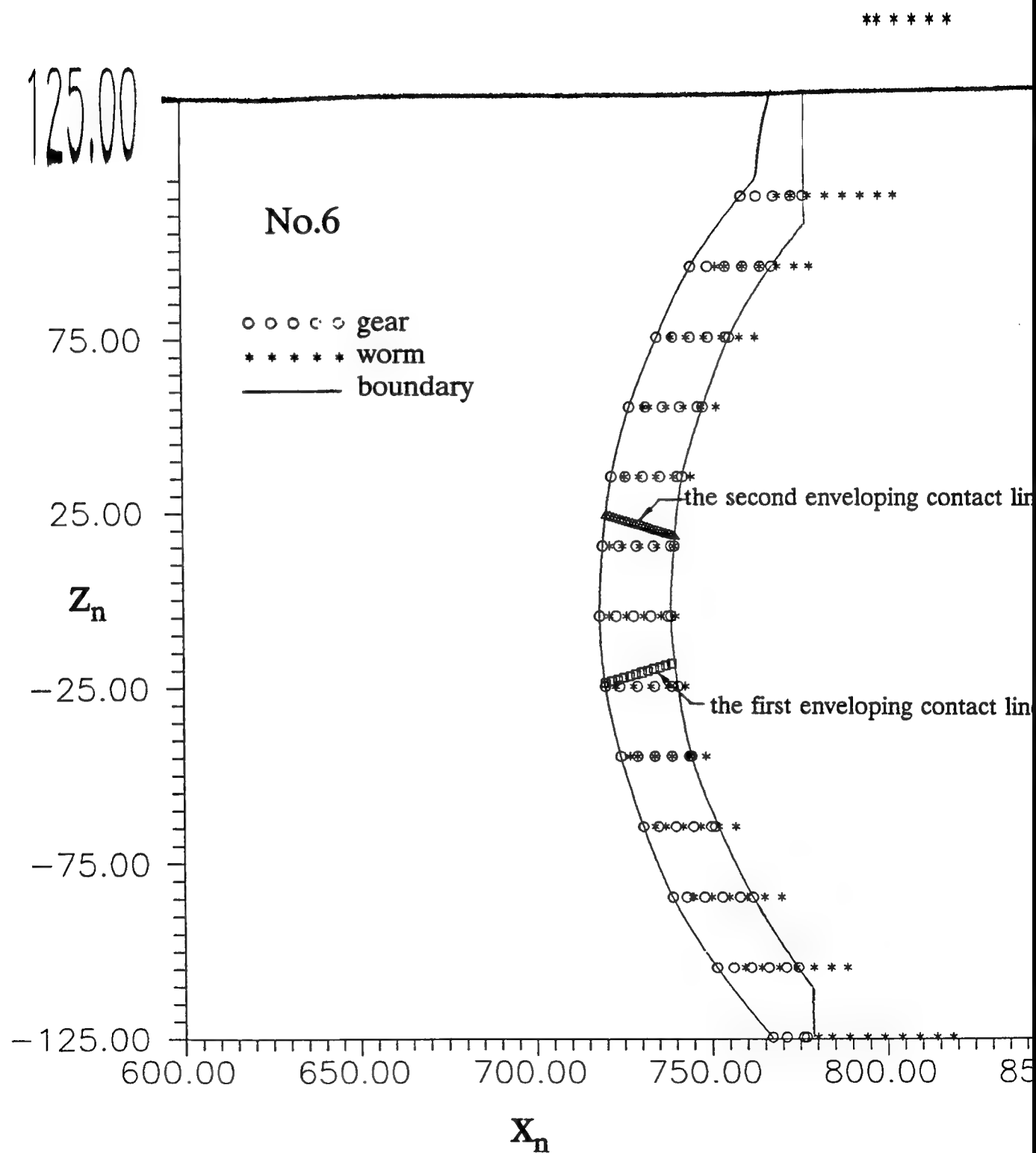
Fig. 3.7   Contact lines on the meshing gear teeth (c)
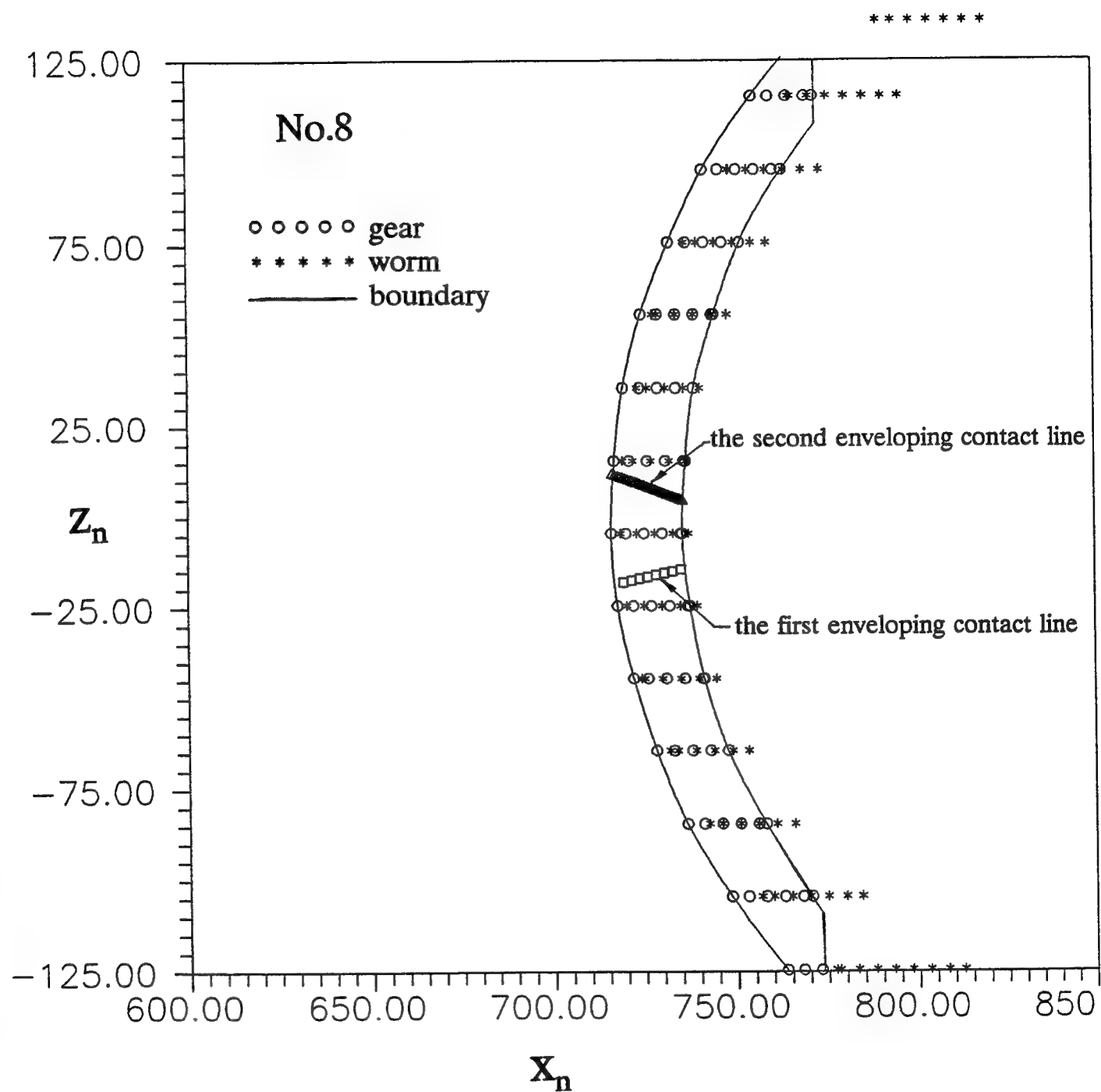
Fig. 3.7   Contact lines on the meshing gear teeth (d)
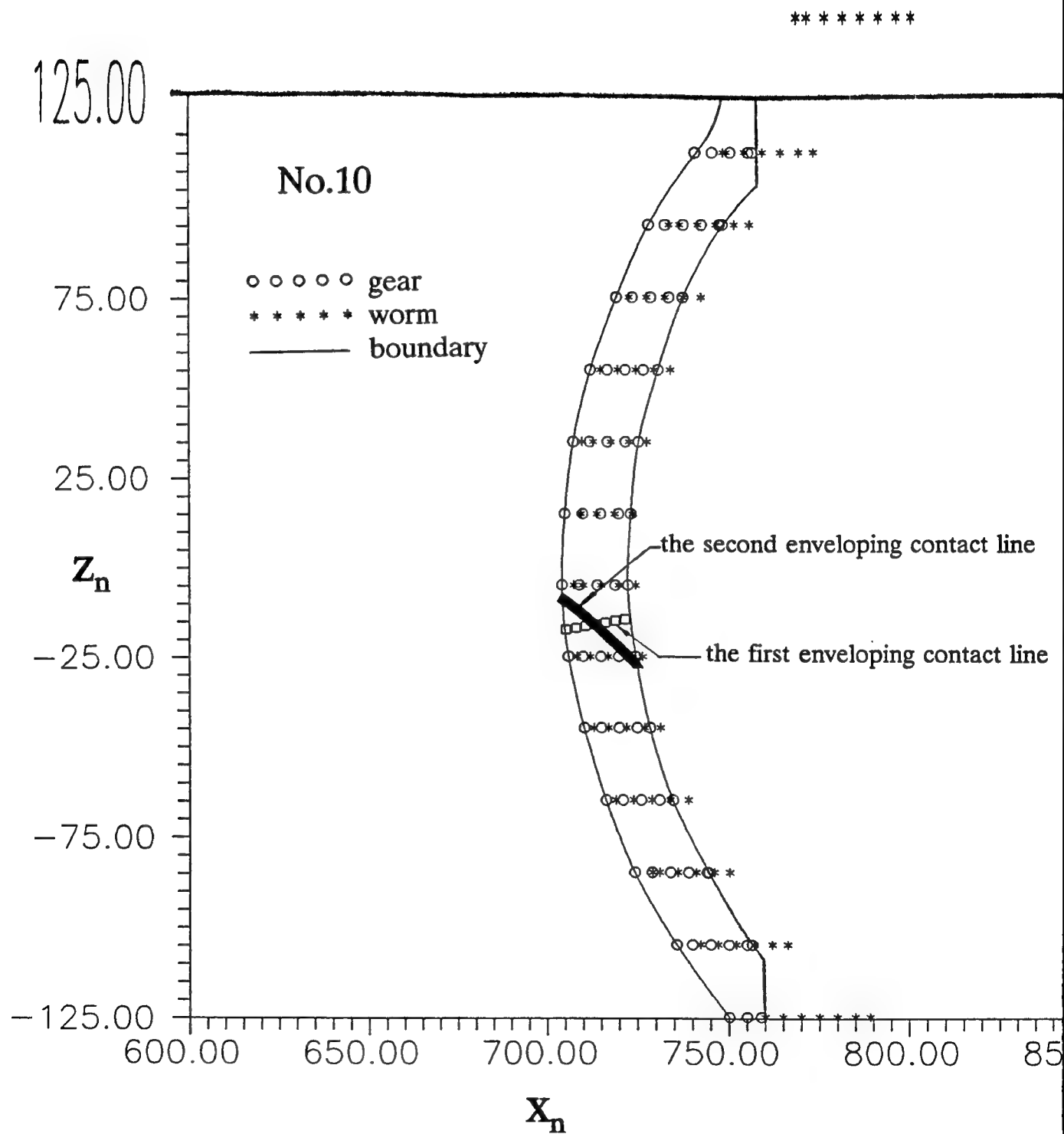
Fig. 3.7   Contact lines on the meshing gear teeth (e)
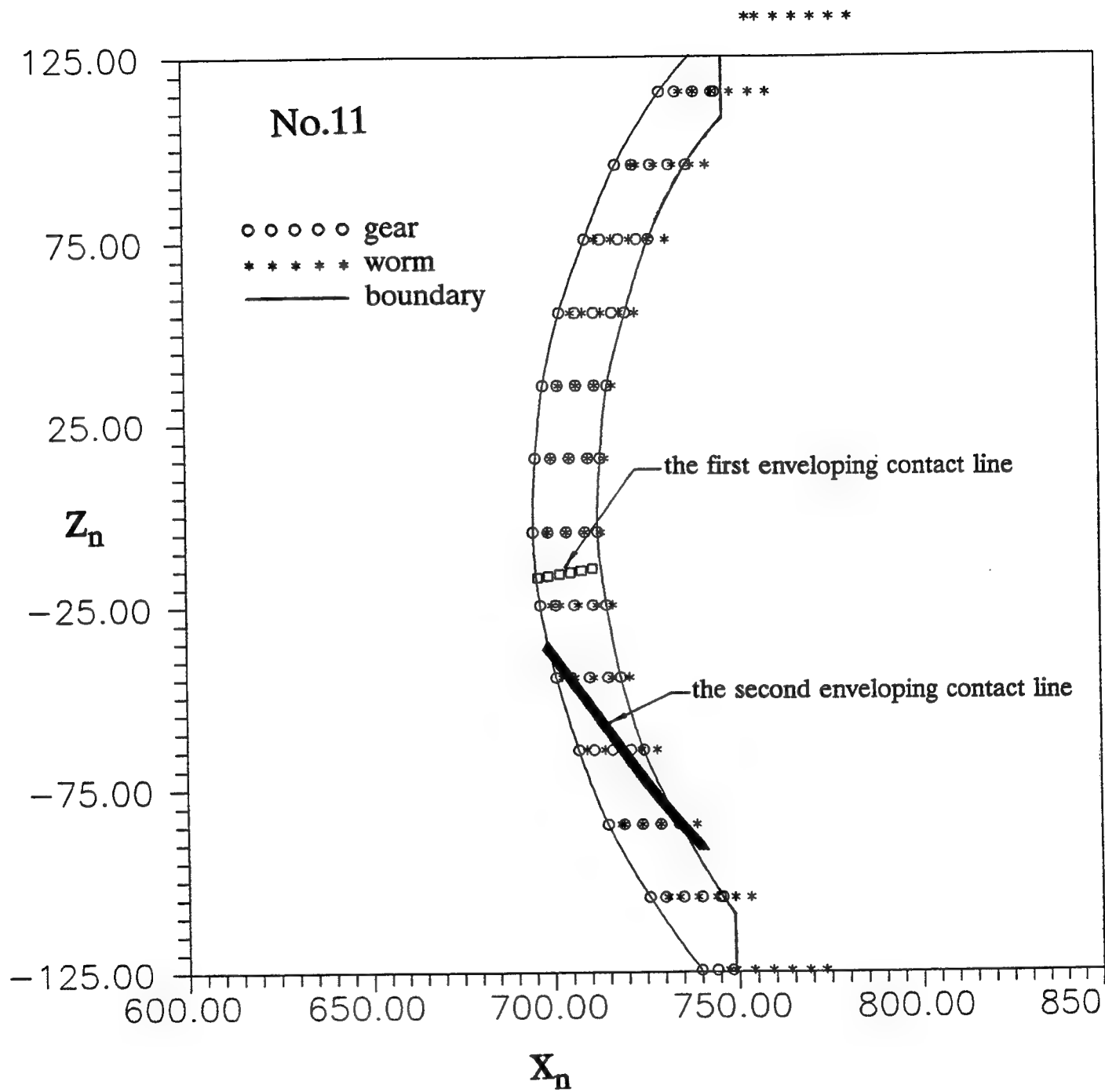
Fig. 3.7   Contact lines on the meshing gear teeth (f)
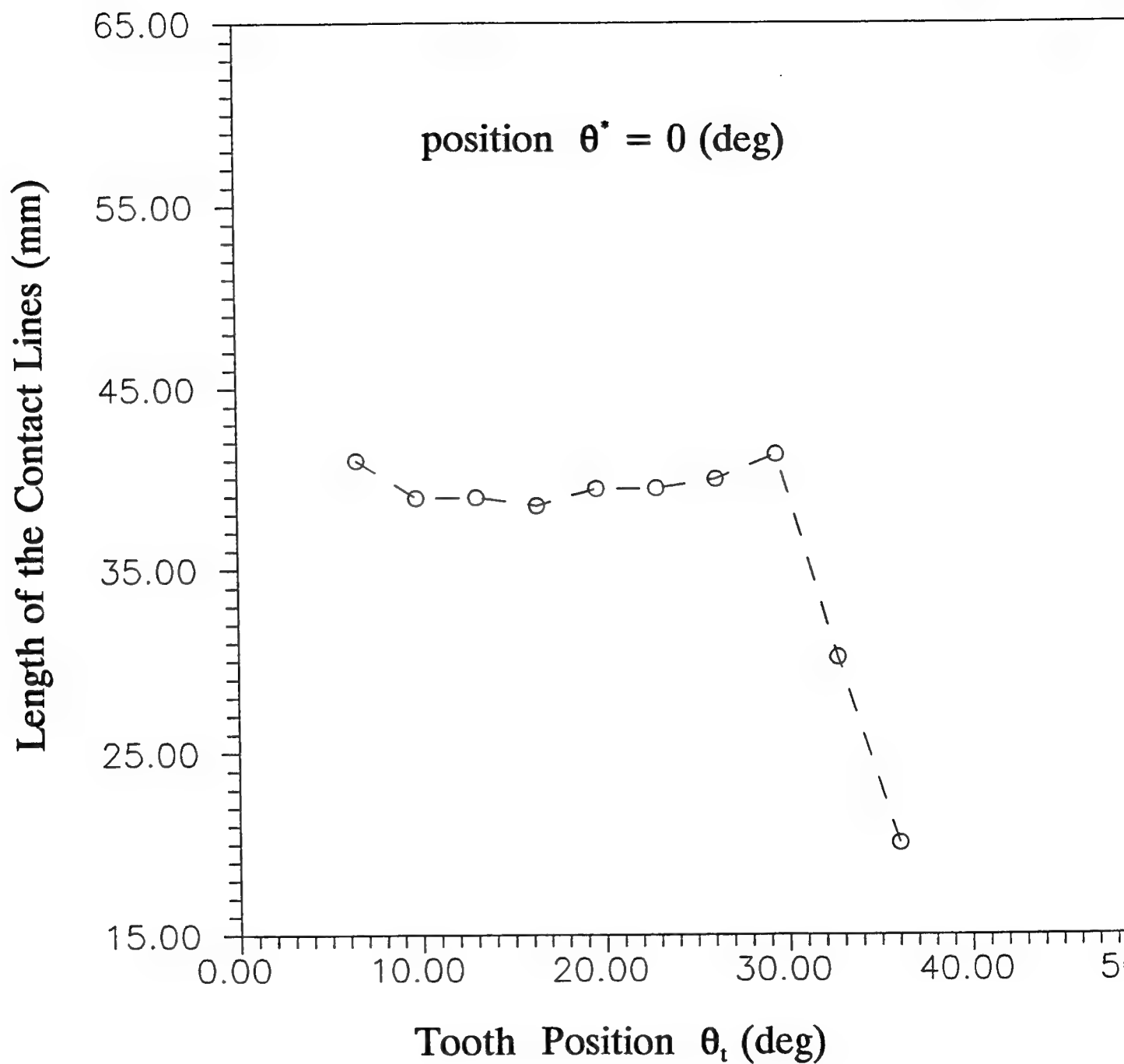
Fig. 3.8 Length of contact lines on the meshing teeth
(a) position of $\theta^* = 0$ deg
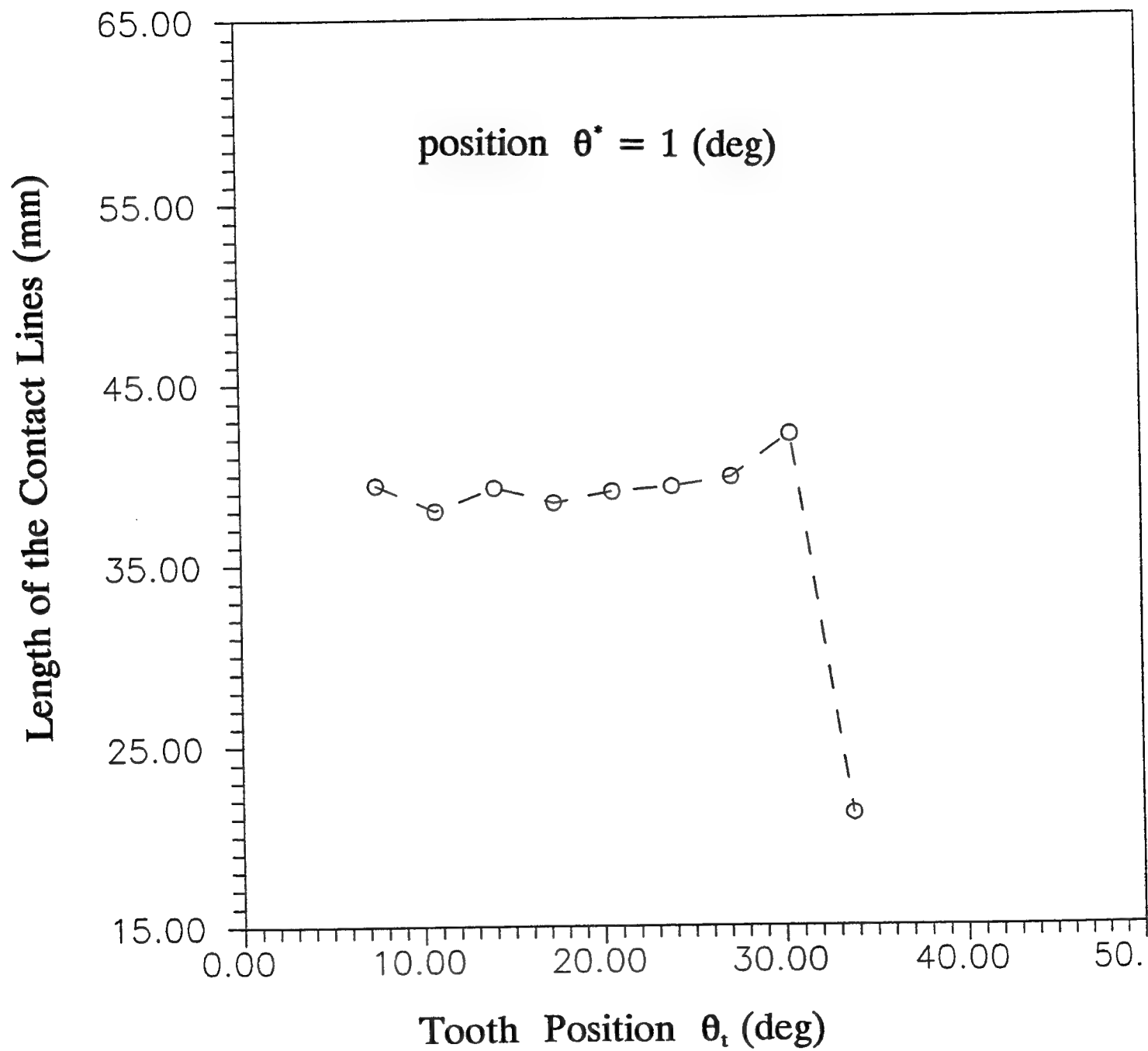
Fig. 3.8   Length of contact lines on the meshing teeth
(b)   position of $\theta^* = 1$ deg
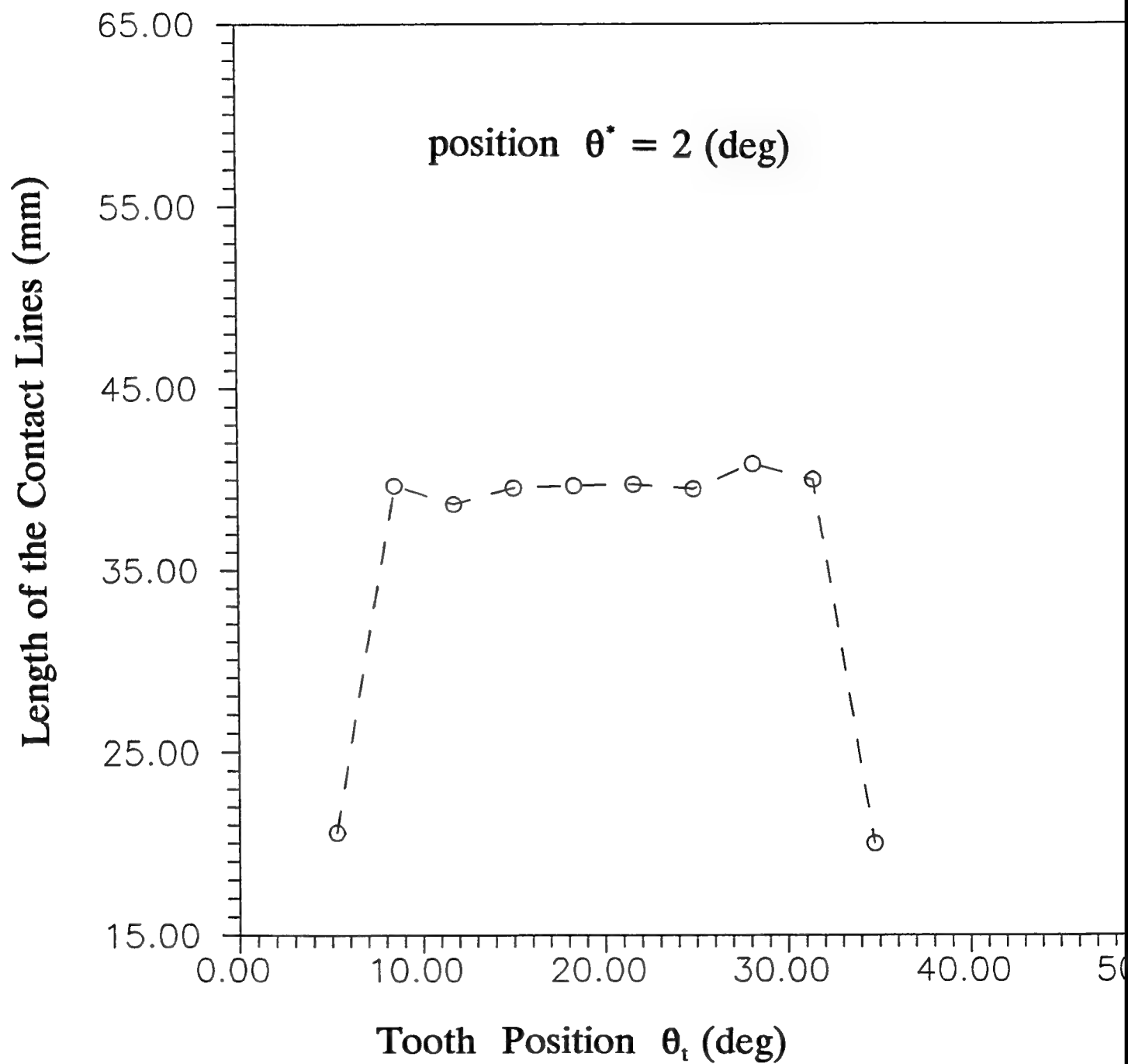
Fig. 3.8   Length of contact lines on the meshing teeth
(c) position of θ^*=2 deg
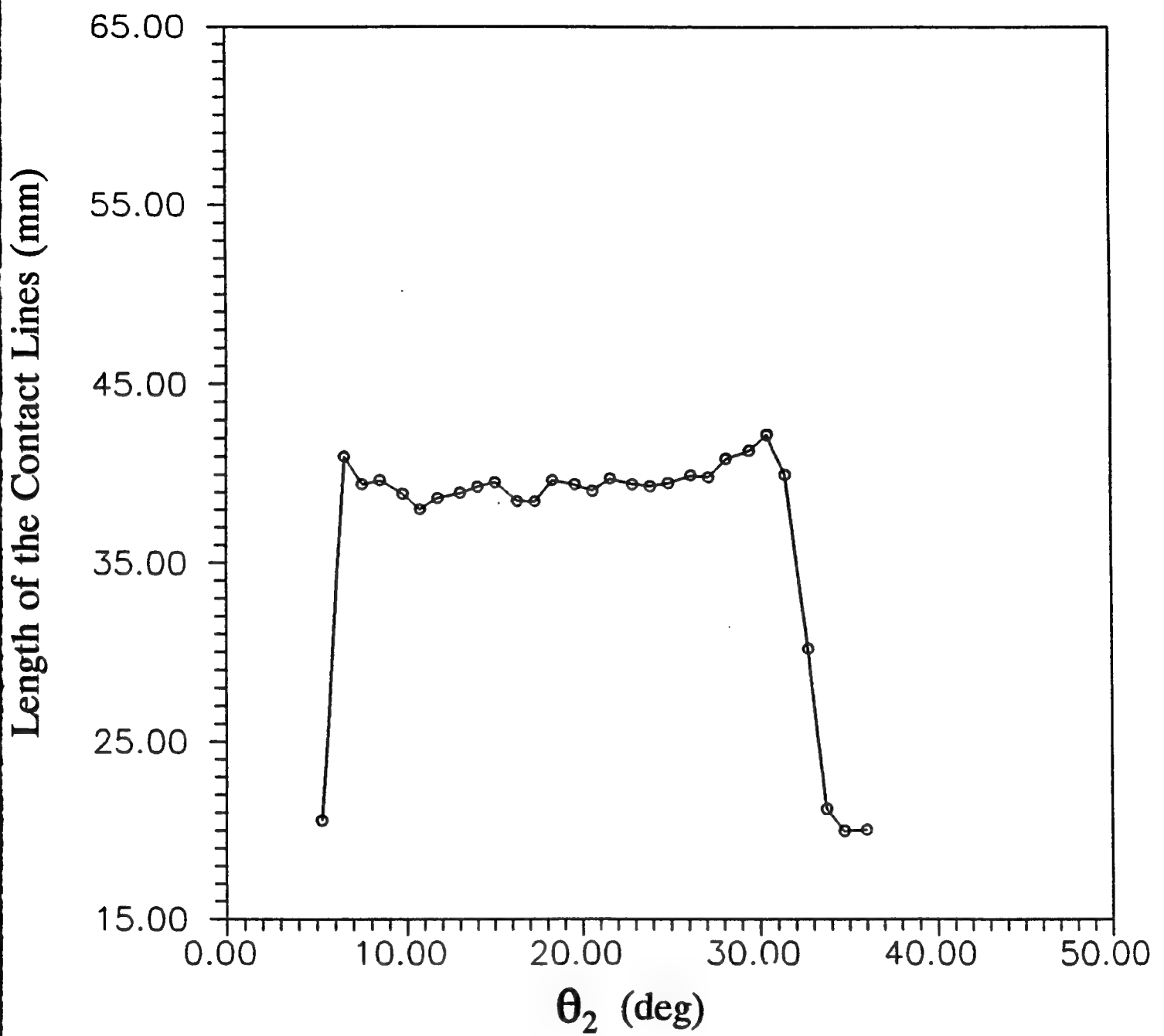
Fig. 3.9   Variation of contact line length on a gear tooth
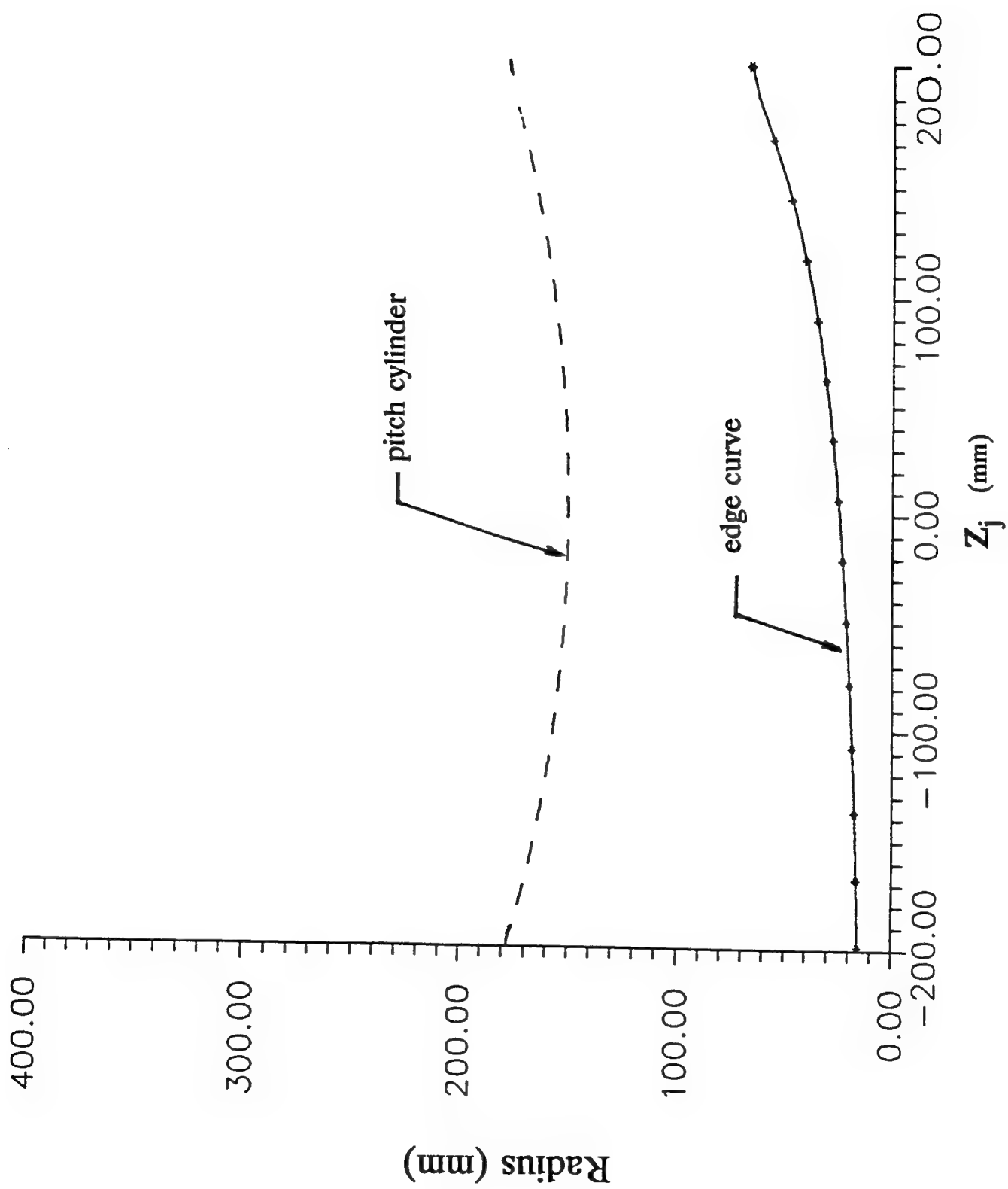
Fig. 3.10 Radial position of the edge curve
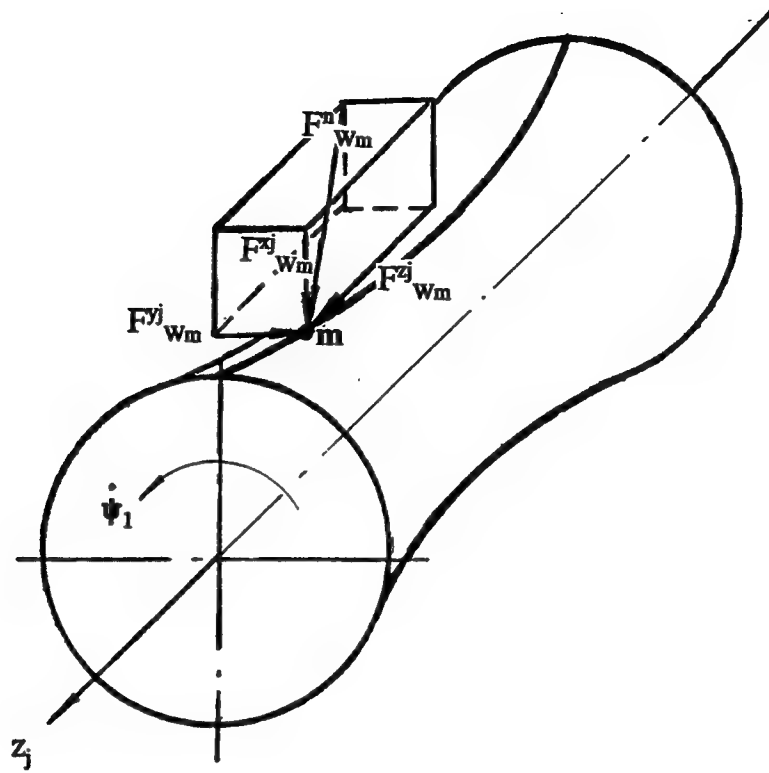
Fig. 4.1 Normal force and its components

71
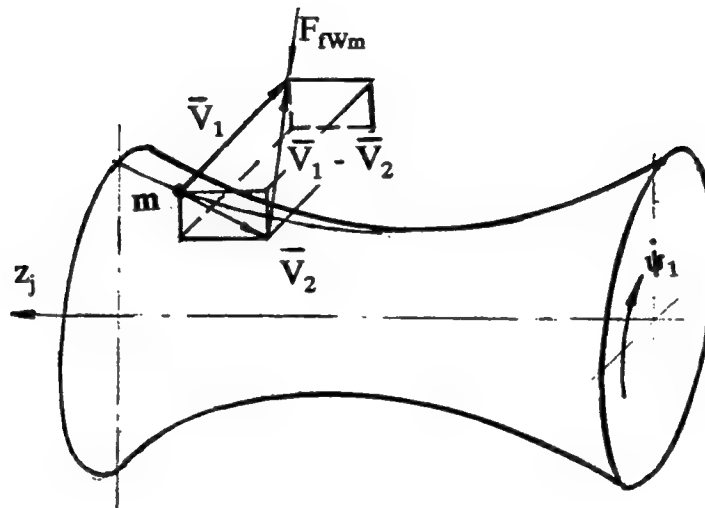
Fig. 4.2 Friction force and its components
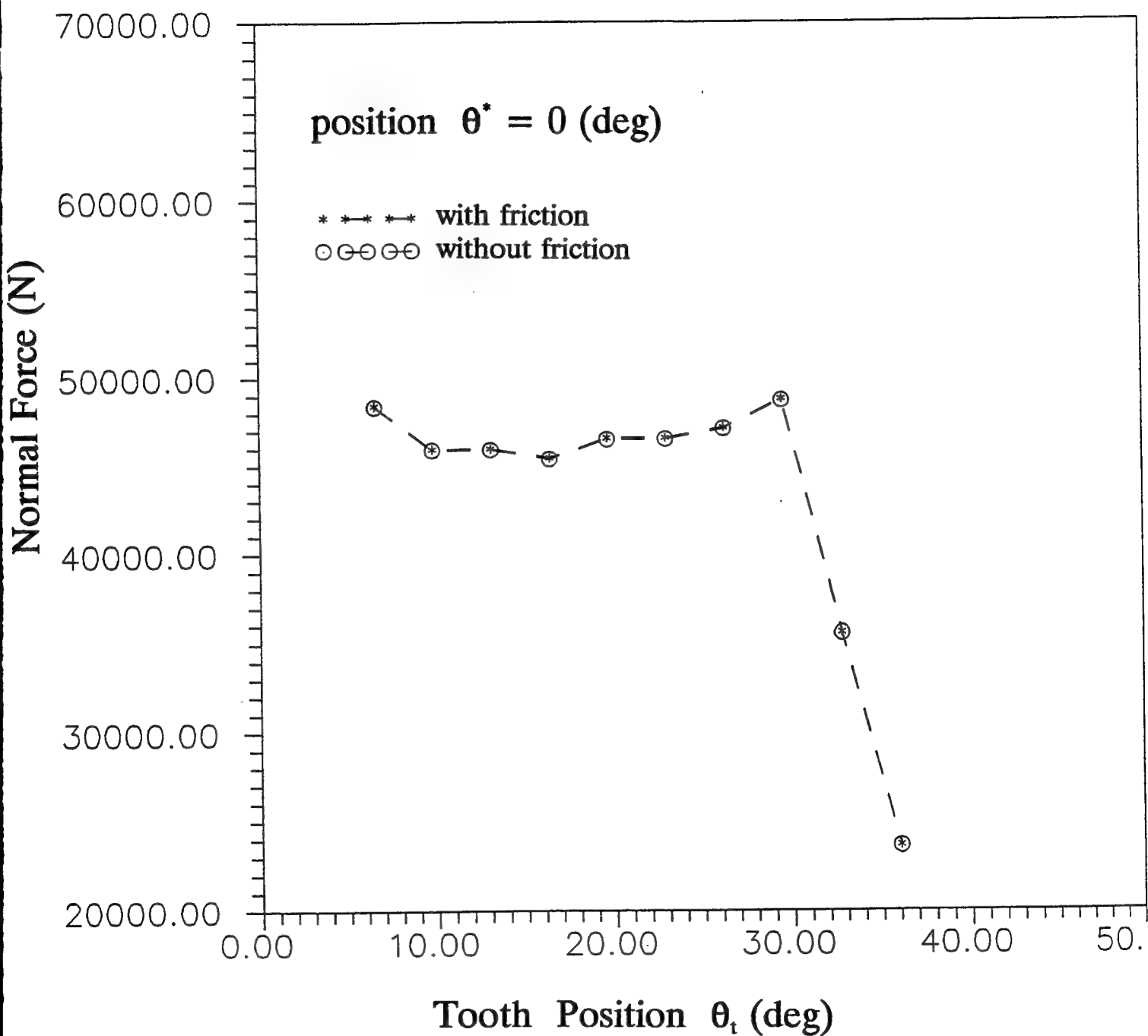
Fig. 4.3   Normal forces on the gear teeth
(a) position of θ^*=0 deg

73
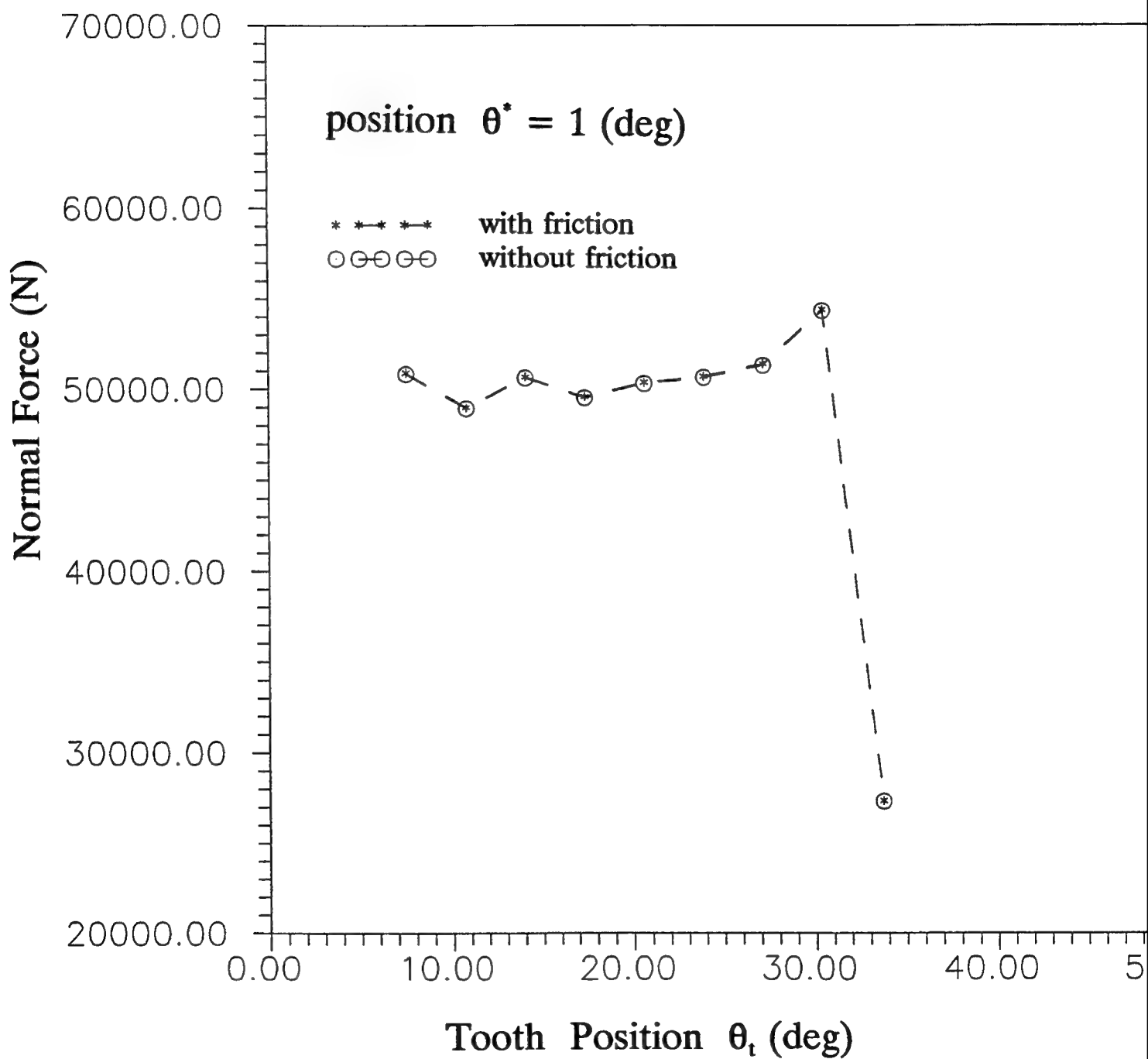
Fig. 4.3   Normal forces on the gear teeth
(b) position of $\theta^* = 1$ deg

74
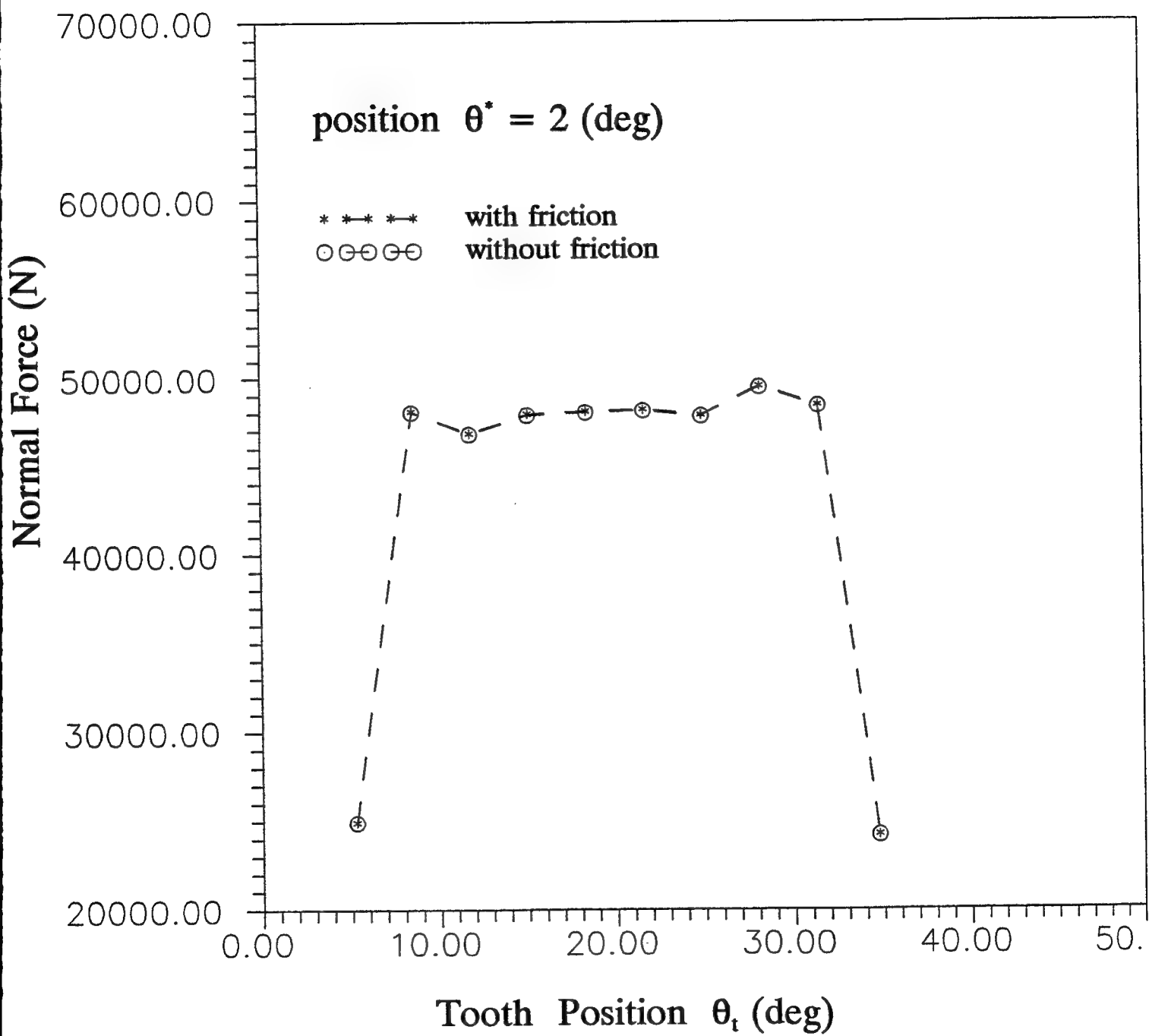
Fig. 4.3   Normal forces on the gear teeth
(c) position of θ^*=2 deg
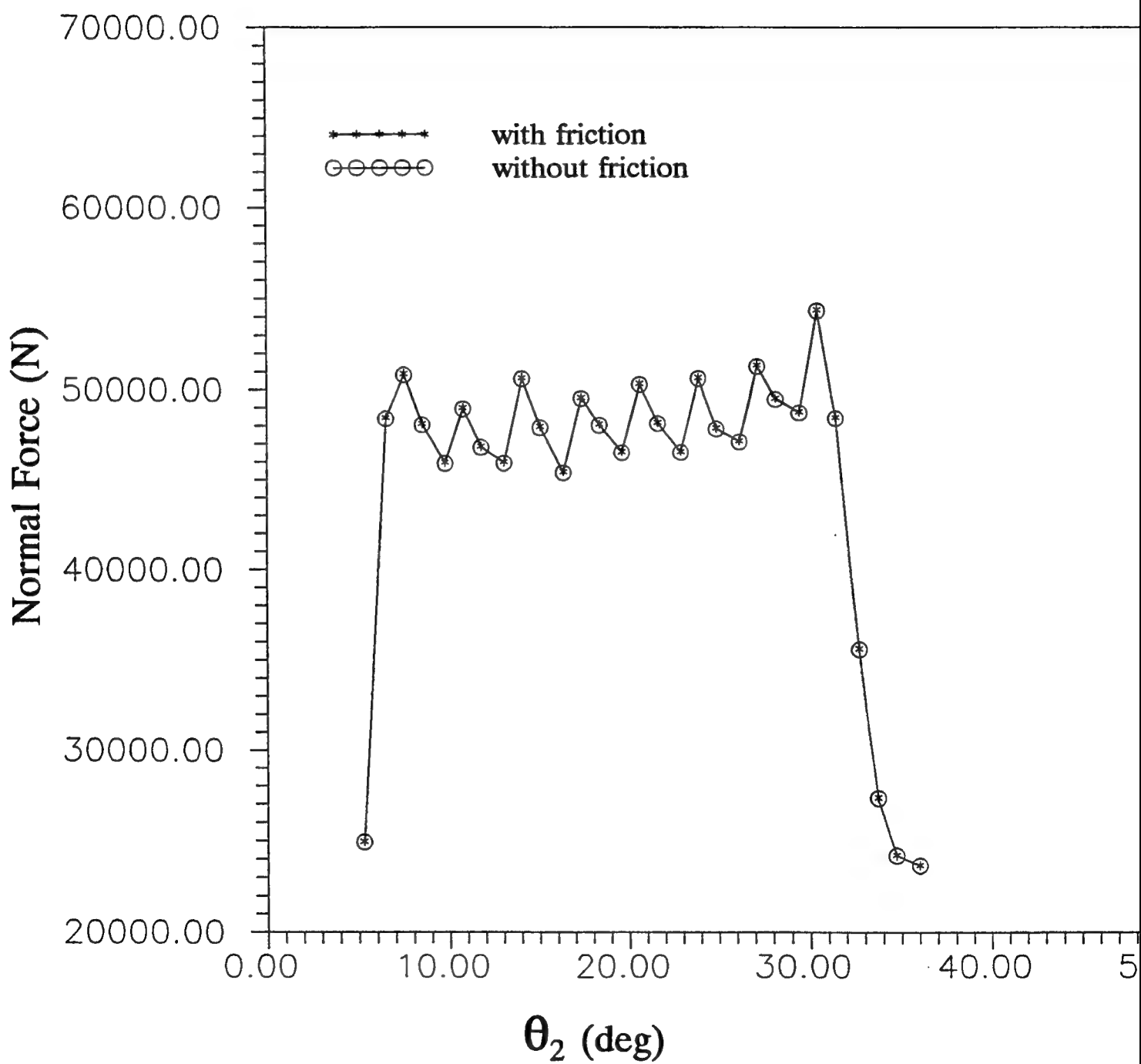
Fig. 4.4 Variation of normal force on a gear tooth
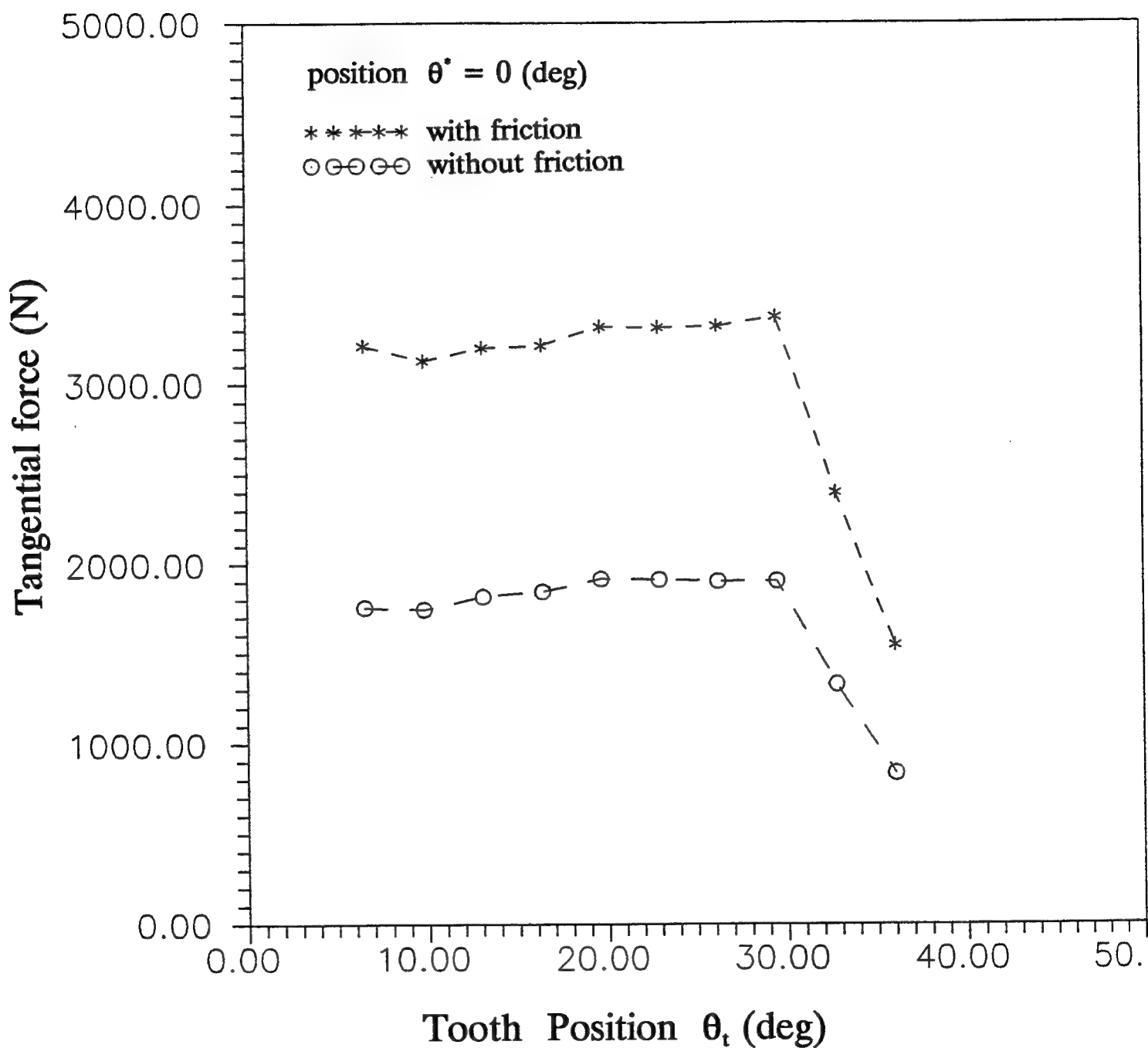
Fig. 4.5   Tangential forces on the worm
(a)   position of $\theta^* = 0$ deg
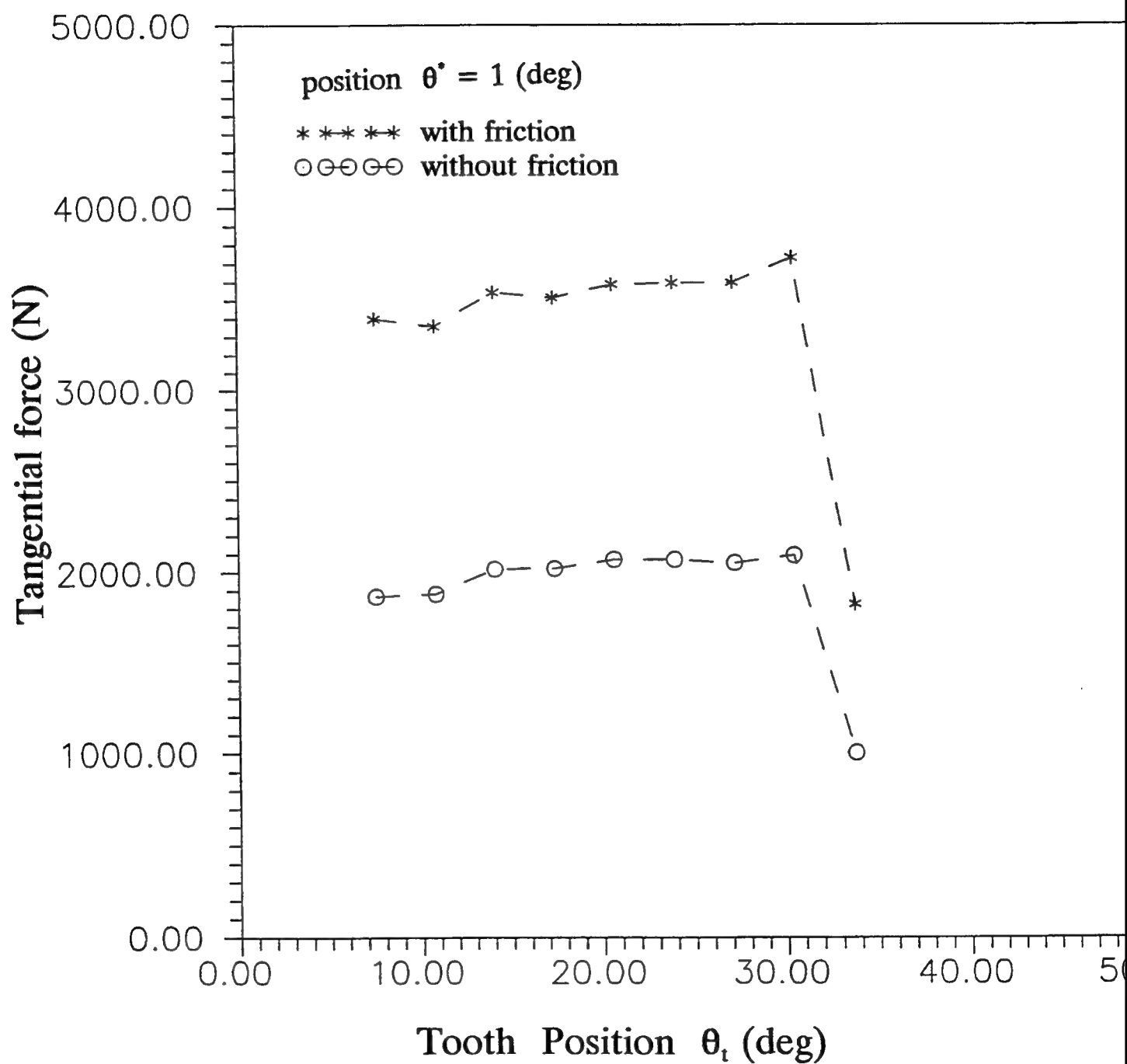
Fig. 4.5   Tangential forces on the worm
(b)   position of $\theta^*=1$ deg
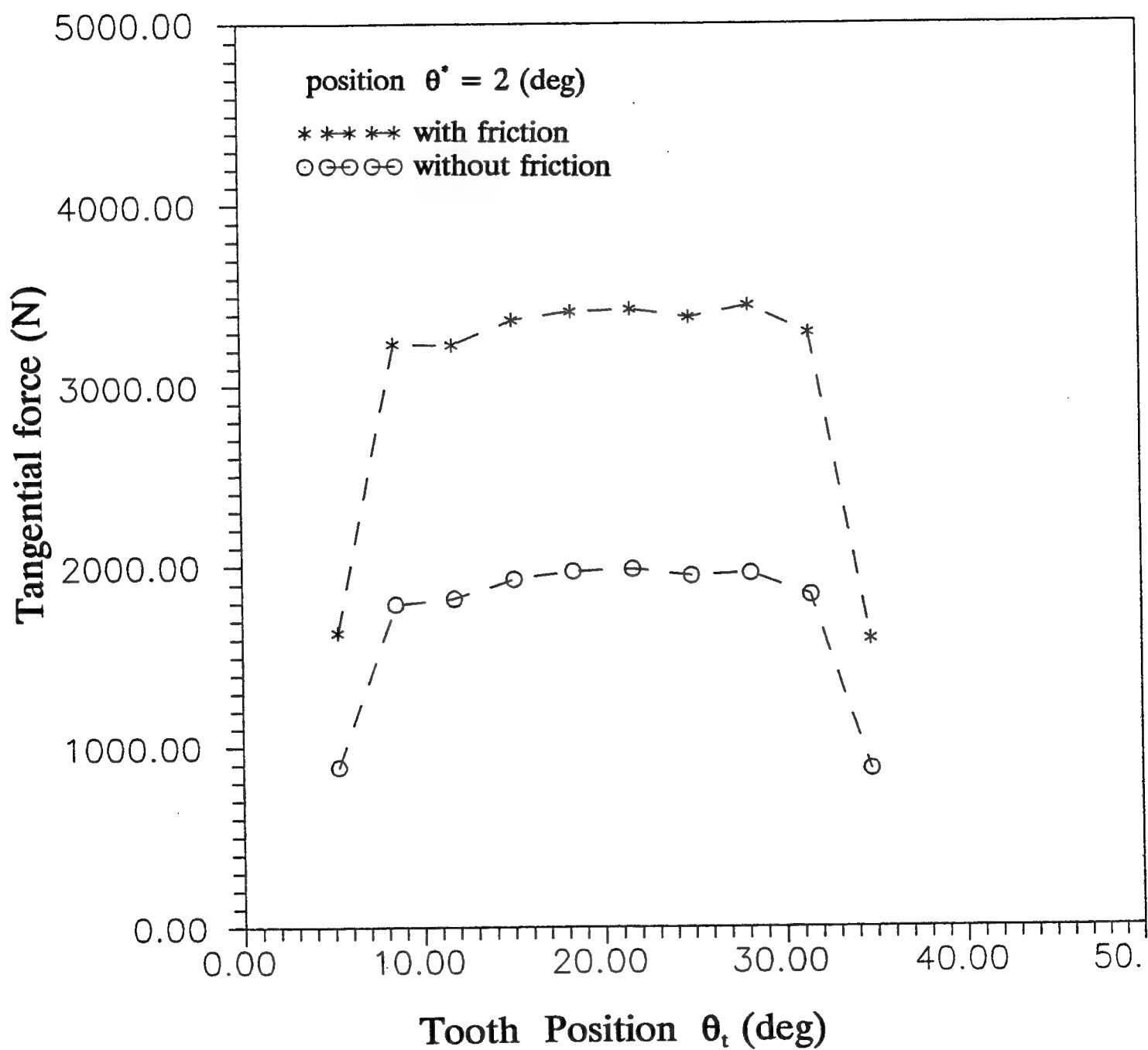
Fig. 4.5   Tangential forces on the worm
(c)   position of $\theta^* = 2$ deg
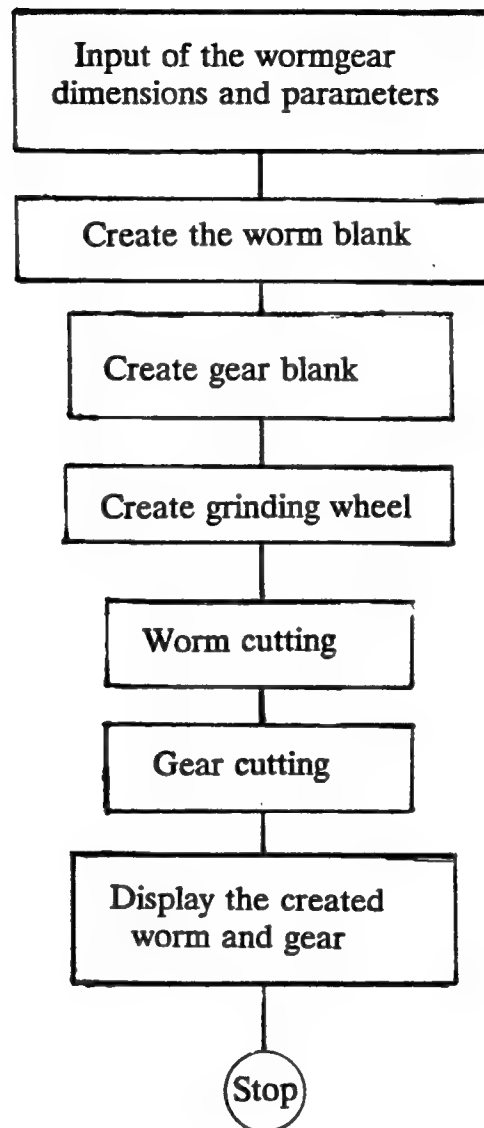
Fig. 5.1 Structure of graphics program
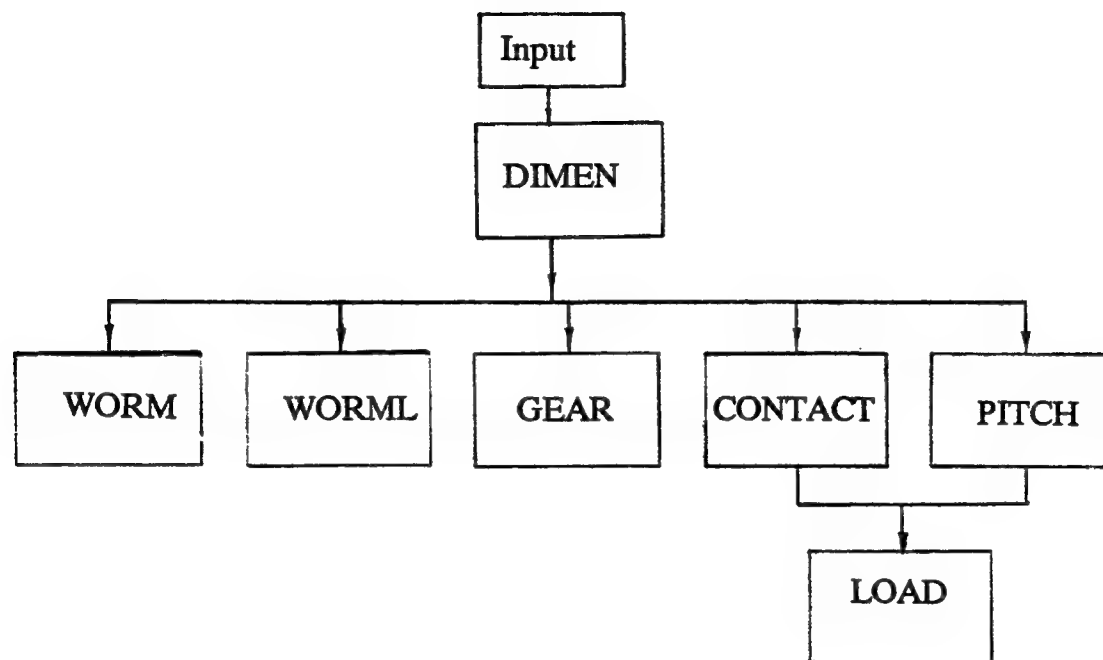
Fig. 5.2 Structure of analyses program

MAIN
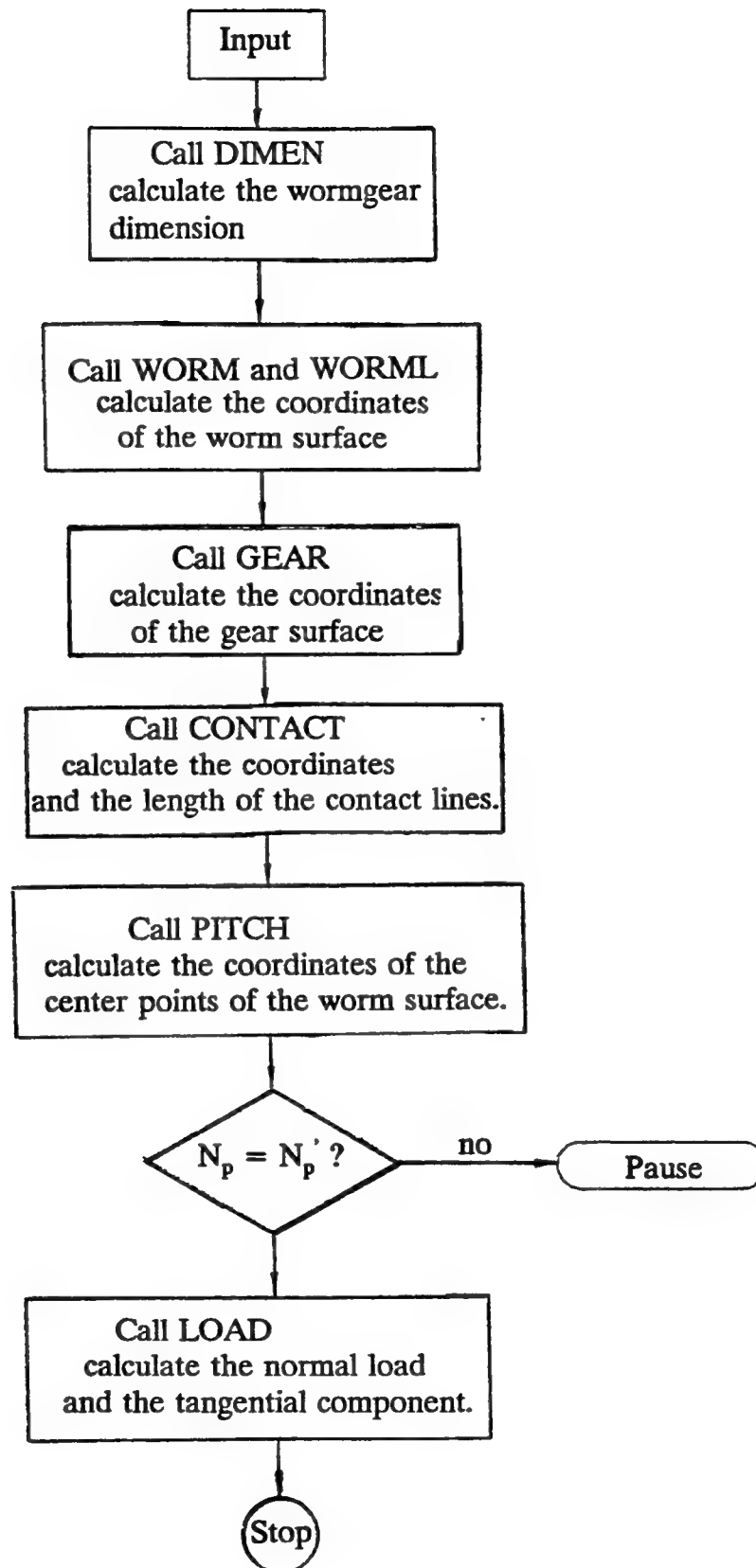
Fig. 5.3 Flow chart of main program

the other side of the worm tooth

one side of the worm tooth
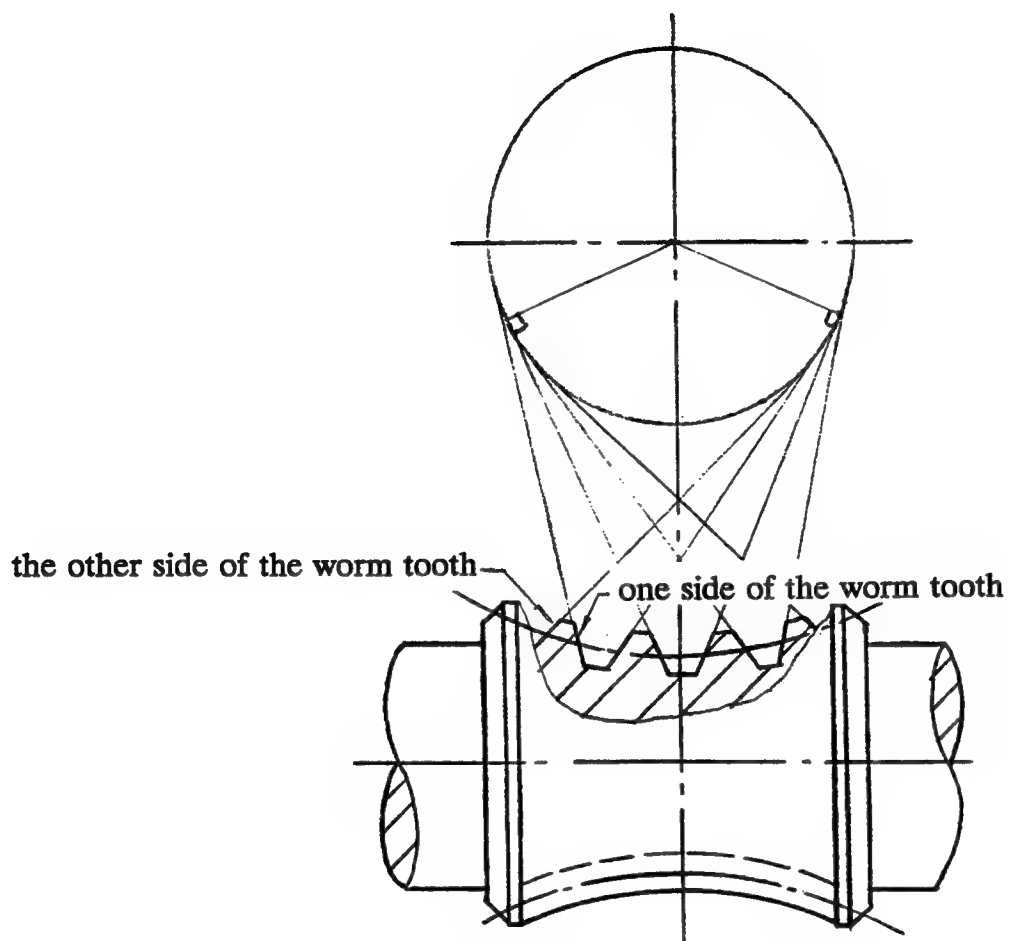
Fig. 5.4   Two sides of the worm thread

83
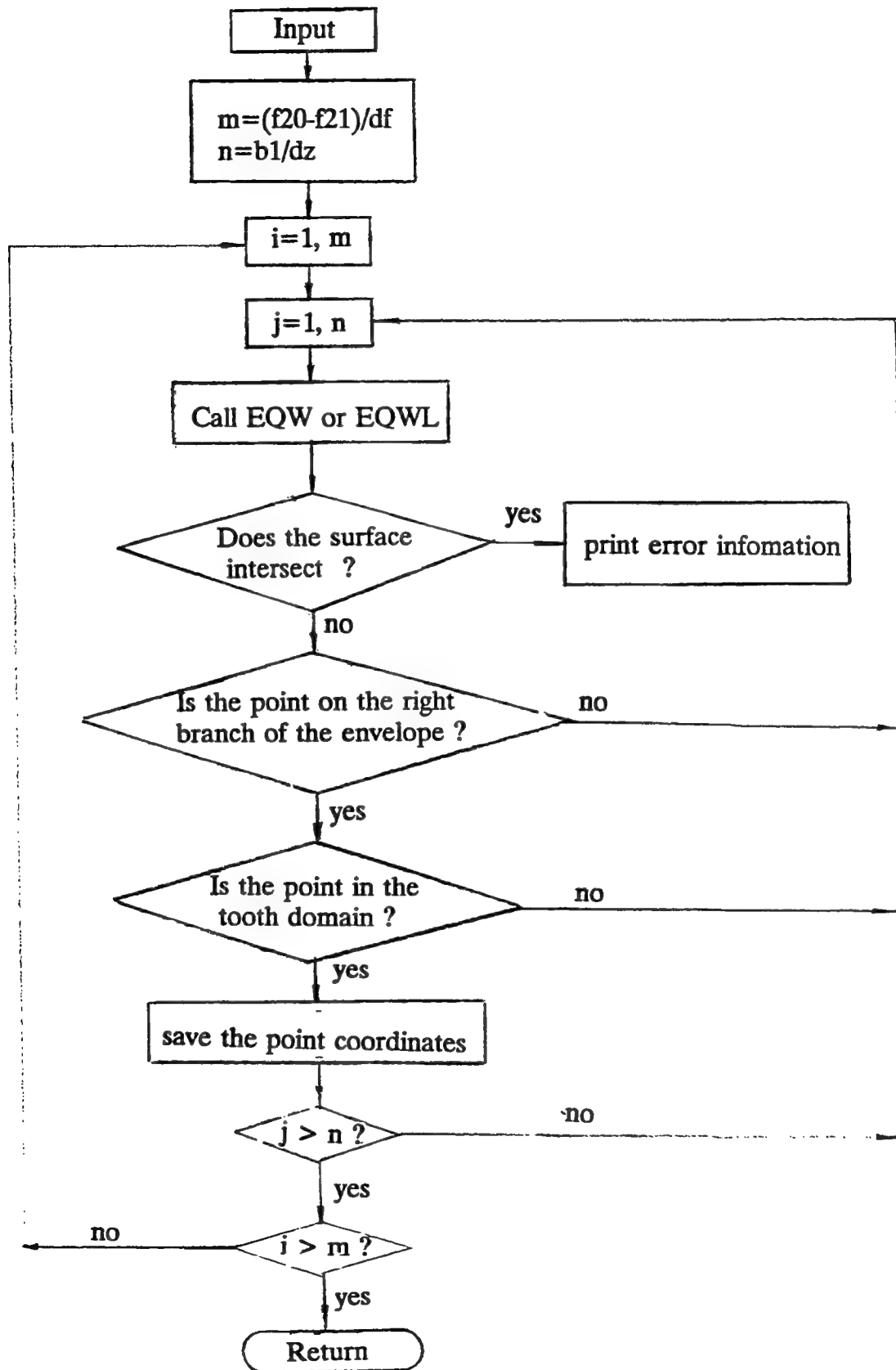
Fig. 5.5 Flow chart of subroutine WORM (also WORML)
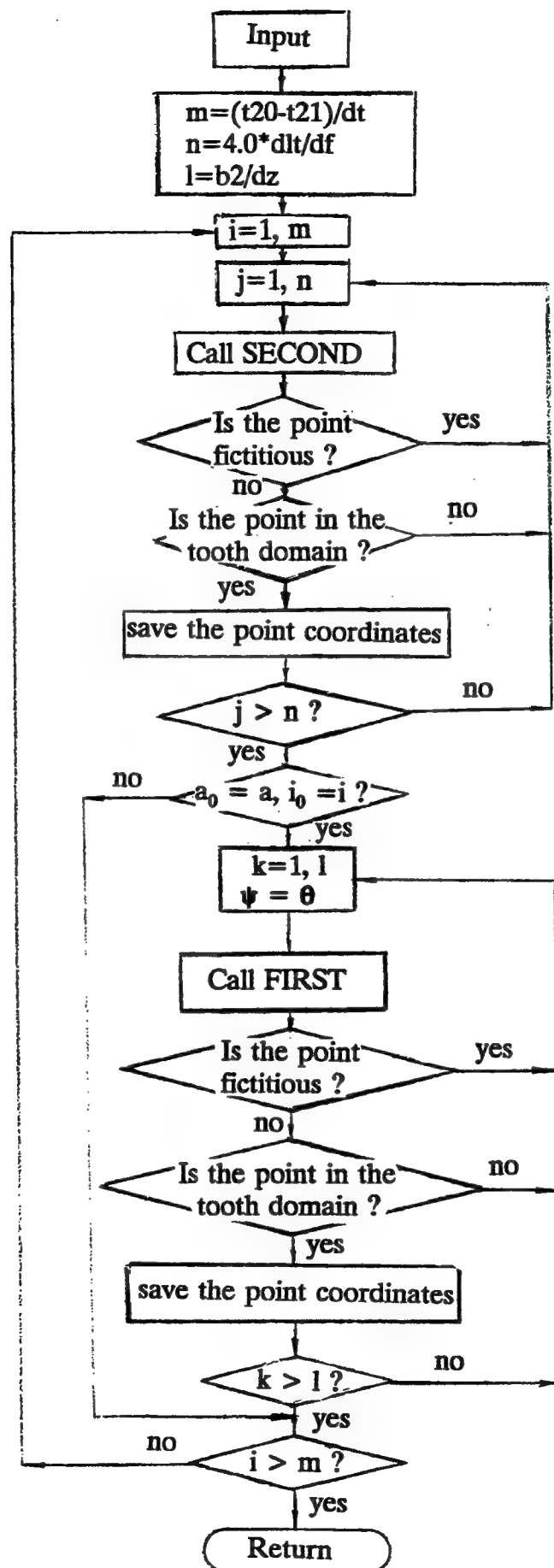
Fig. 5.6 Flow chart of subroutine GEAR

85

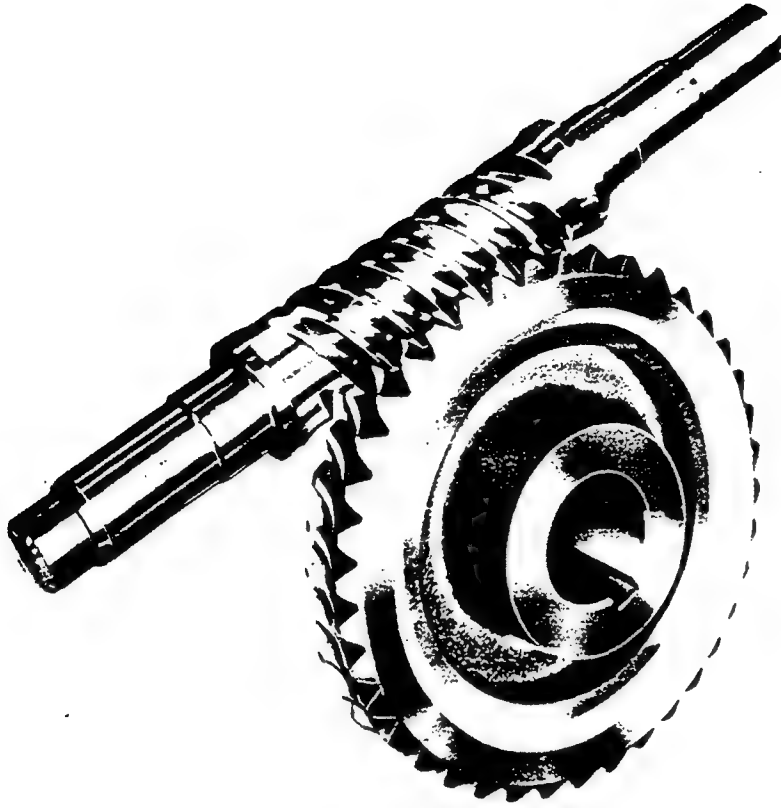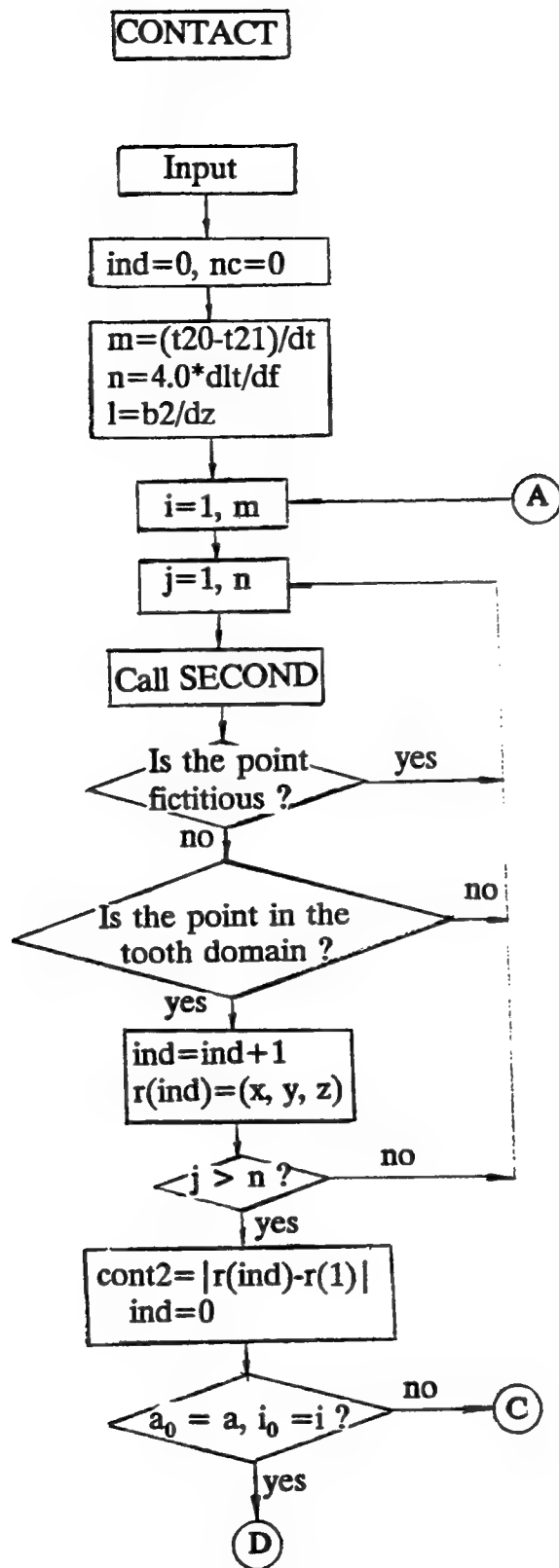Fig. 5.7   Flow chart of subroutine CONTACT

Fig. 5.7   (continued)

87
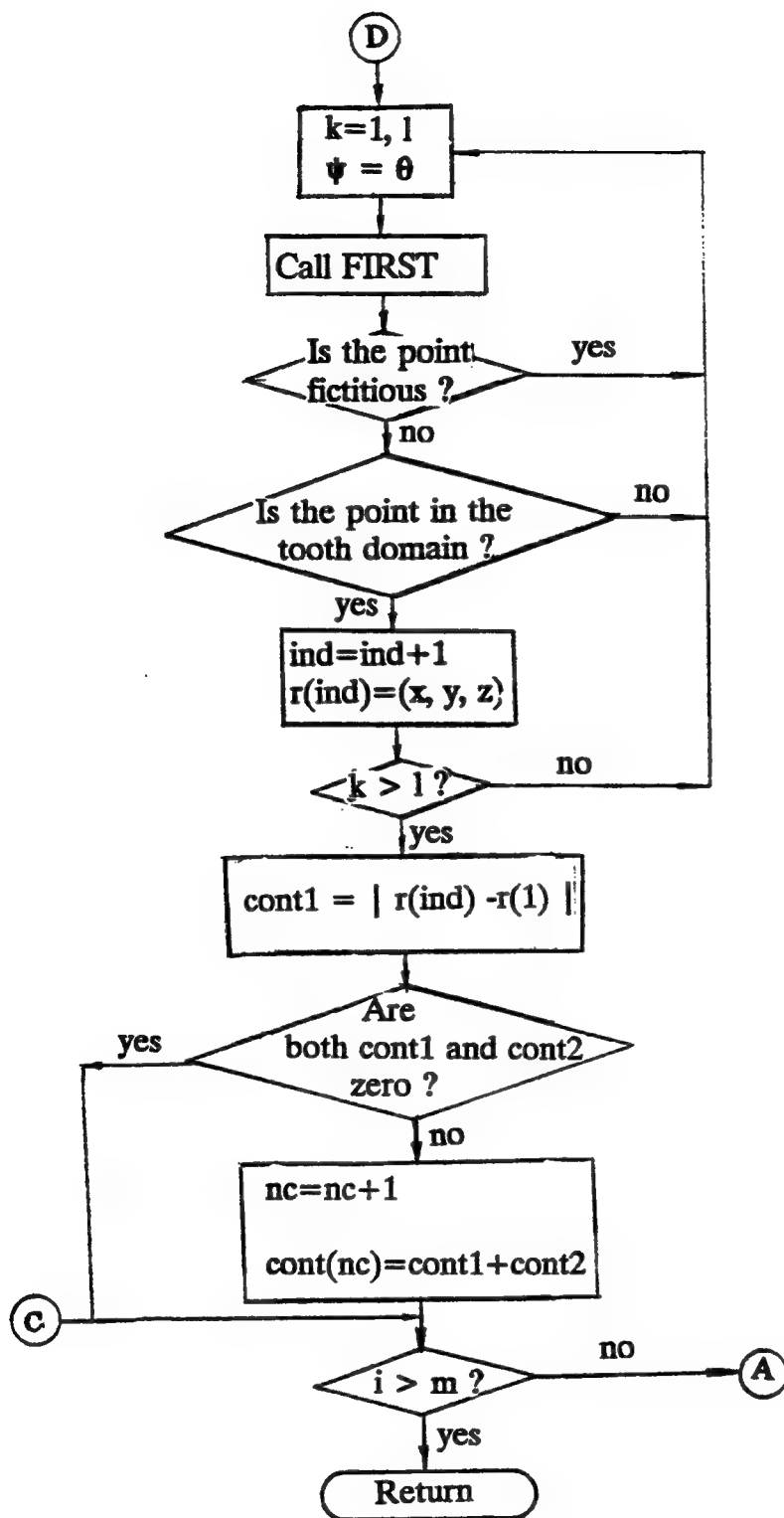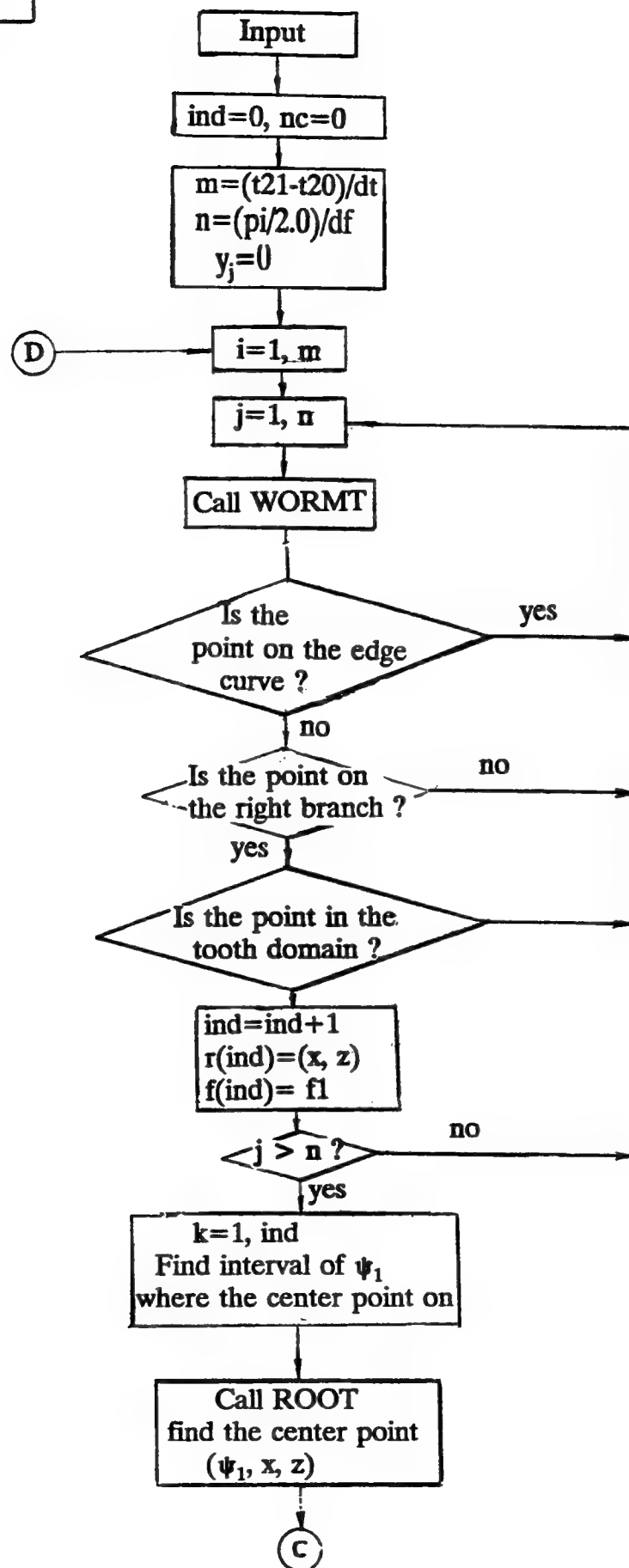
Fig. 5.8   Flow chart of subroutine PITCH
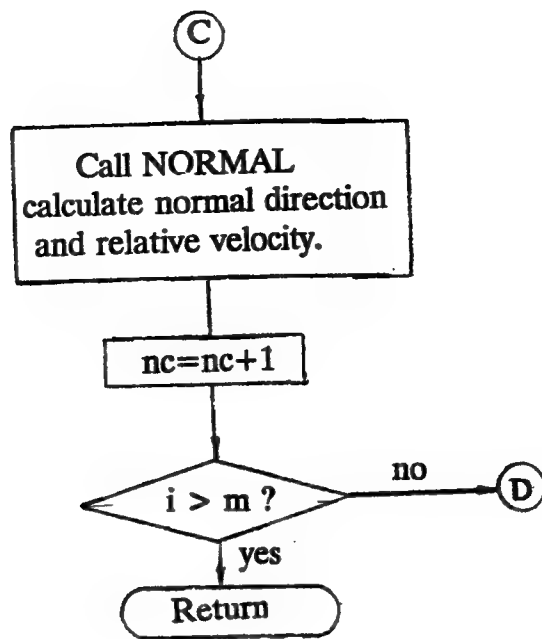
Fig. 5.8   (continued)

Appendix 1 Graphics Program (NASA.PRG)

The procedure to execute 'NASA.PRG'

(1)   From the SUNY-Binghamton operating system prompt (assumed here to be $):

    $ ready: type 'runideas' <return>

    $ jump to IBM screen (use Alt-jump)

    $ user i.d.: type 'ideas2'

    $ password: type 'no'

    $ jump back to BINGVMB screen type 'none' (use Alt-jump)

    $ disk space: type a number such as '15000'

       (waiting for temporary disk to be created)

    $ user i.d.: type your own i.d. number and read password

    $ terminal type: type 'i5080'

    $ jump screen (use Alt-jump)

       (waiting for I-DEAS screen to appear)

(2)   Under the I-DEAS screen, from the I-DEAS operating prompt (assumed here
to be #):

    Enter model file name (none)# <return>

    New model file? (yes)# <return>

    Enter model description (none)# <return>

    Enter system of units (SI)# <return>

    Enter operation mode (Normal_Operation)# <return>

(3)   Then choose which "family" of I-DEAS you want. Notice that choices are
in the form of a 'menu'. So you can pick a choice with the mouse or type
under the I-DEAS operating prompt (assumed here to be #):

    Select menu# SO <return>

    (waiting for Solid Modeling Screen to appear)

    Select menu# CG <return>

90

Select menu# **MF** <return>

        Select menu# **PR** <return>

        Select menu# **R** <return>

        File name# **NASA** <return>

Then the program is run interactively. All the results will be seen from the

screen.

(4)   The created object can be observed again before exiting I-DEAS system.

The procedure to observe the created object is:

        Select menu# **MA** <return>

        Select menu# **G** <return>

        Enter object name# type object name (worm or gear)

        (the created object appears on the screen)

Once the created object is obtained, choose from the menu options such as

orient, view, display and AU (auto scale) to observe the object.

(5)   When the observation is completed, put the created object away from the

screen by:

        Select menu# **/** <return>

        Select menu# **MA** <return>

        Select menu# **PU** <return>

(6)   When all the observations are completed, if needed the created object

can be saved by:

        Select menu# **/** <return>

        Select menu# **MA** <return>

        Select menu# **WR** <return>

        Select menu# **U** <return>

        Enter the file name# type the name you choose <return>

(7)   Exit I-DEAS screen by:

     Select menu# / <return>

     Select menu# type 'exit' <return>

Program Listing

```
C :
C :   *****************************************************************
C :     This is an ideas program to construct  a wormgear set by the
C :     principle of enevlop.  The constructing process is same as
C :     the manufacturing process exactly. To cut four worm threads
C :     the 'wn' must be equal to 360/dtw and 'gn' must be 4*360/dtg.
C :
C :     The basic parameters provided are :
C :
C :       a0-----center distance of cutting      (a0=880.0 mm)
C :       db-----base circle diameter            (db=519.3182 mm)
C :       b1-----face width of worm              (b1=383.0 mm)
C :       b2-----face width of gear              (b2=249.0 mm)
C :       d2-----pitch diameter of gear          (d2=1460.00 mm)
C :       da2----throad diameter of gear         (da2=1478.7455 mm)
C :       df2----root diameter of gear           (df2=1437.0888 mm)
C :       df1----root diameter of worm           (df1=277.0888 mm)
C :       ratio--speed ratio of wormgear         (ratio=110)
C :       n2-----number of gear teeth            (n2=110)
C :       dt1----angle of worm turning in each cut
C :       dt2----angle of gear turning in each cut
C :       wn-----times of worm cutting
C :       gn-----times of gear cutting
C :       bta----apex angle of generating plane (bta=10 deg)
C :       psif---angle of meshing start         (psif=5.209 deg)
C :       reps---angle of two cutting surfaces  (reps=39.8724 deg)
C :             ** 'reps' is calculated by the following formular:
C :                 reps=180-{2*[90-arcsin(db/d2)]+s2/r2} , here
C :                 s2=0.55*pi*d2/n2  i.e.  s2/r2=1.1*180/n2 (deg)
C :   *****************************************************************
C :
C :
C :     =================================
C :     =  basic parameters input  =
C :     =================================
K :   "/"
K :   #echo list
K :   #a0=880.0
K :   #n2=110
K :   #ratio=110
K :   #db=519.3182
K :   #bta=5.00
K :   #d1=300.0
K :   #d2=1460.0
K :   #da1=318.7455
K :   #da2=1478.7455
K :   #df1=277.0888
K :   #df2=1437.0888
K :   #b1=393.0
K :   #b2=249.0
C :
K :   #rb=db/2.0
K :   #r1=d1/2.0
K :   #r2=d2/2.0
K :   #ra1=da1/2.0
K :   #ra2=da2/2.0
K :   #rf1=df1/2.0
K :   #rf2=df2/2.0
K :   #dh=2.0829
C :
```

```
K :  #psif=5.209
K :  #reps=39.8724
K :  #aa=a0-sqrt((rf2+dh)**2-b1*b1/4.0)
K :  #bb=sqrt(ra2*ra2-rb*rb)
K :  #cc=a0-sqrt(r1*r1-b2*b2/4.0)
K :  #dd=a0-sqrt(ra2*ra2-b1*b1/4.0)
K :  #ee=a0-ra2
K :  #dtw=8.00
K :  #dtg=15.0
K :  #wn=5
K :  #gn=3
C :
C :          ==================================
C :          = creat 2-D profile of worm blank =
C :          ==================================
C :
K :  "CR"
K :  "PR"
K :  "K"
K :  0,r2
K :  a0-cc,r2
K :  "F"
K :  a0-cc,b1/2.0
K :  "F"
K :  aa,b1/2.0
K :  "F"
K :  a0-rf2-dh,0
K :  aa,-b1/2.0
K :  "F"
K :  a0-cc,-b1/2.0
K :  "F"
K :  a0-cc,-r2
K :  "F"
K :  0,-r2
K :  "F"
K :  "C"
K :  "/"
K :  "MA"
K :  "PR"
K :  "STO"
K :  p1
K :  "/"
K :  "DE"
K :  "P"
K :  "Y"
K :  "/"
C :
C :          ======================================
C :          = creat 2-D profile of worm root blank =
C :          ======================================
C :
K :  "CR"
K :  "PR"
K :  "K"
K :  0,b1/2
K :  dd,b1/2
K :  "F"
K :  ee,0
K :  dd,-b1/2.0
K :  "F"
K :  0,-b1/2.0
K :  "F"
```

94

```
K : "C"
K : "/"
K : "MA"
K : "PR"
K : "STO"
K : p2
K : "/"
K : "DE"
K : "P"
K : "Y"
K : "/"
C :
C :
C :            =================================
C :            = creat 3-D gear blank  =
C :            =================================
C :
K : "TA"
K : "O"
K : "/"
K : "CR"
K : "DEF"
K : "CY"
K : "F"
K : 32
K : "D"
K : "D"
K : "CY"
K : cc,b2
K : "/"
K : "OR"
K : "RO"
K :
K : 90,0,0
K : "MO"
K : "D"
K : 0,1,0
K : a0
K : "/"
K : "MA"
K : "PU"
K : g2
K :
C :
C :
C :            =================================
C :            = creat a 3-D worm cutter =
C :            =================================
C :
K : "/"
K : "CR"
K : "SPH"
K : "A"
K : "C"
K : 6
K : "D"
K : ra2+2*dh
K : "/"
K : "CO"
K : "P"
K : "T"
K : "K"
K : -rb,0,0
K : -rb,a0,0
```

```
K :  0,0,rb/tan(bta)
K :
K :  "P"
K :  "T"
K :  "K"
K :  rb*cos(reps),-rb*sin(reps),0
K :  rb/cos(reps),0,0
K :  0,0,rb/tan(bta)
K :  "N"
K :  "/"
K :  "OR"
K :  "MO"
K :  "D"
K :  0,1,0
K :  a0
K :  "R"
K :  "K"
K :  0,a0,0
K :  0,0,psif
K :  "/"
K :  "MA"
K :  "STO"
K :  c2
C :
C :
C :                 =========================================
C :                 = creat four connected 3-D worm cutters =
C :                 =========================================
C :
K :  "/"
K :  "OR"
K :  "R"
K :  "K"
K :  0,a0,0
K :  0,0,360/n2
K :  "/"
K :  "MA"
K :  "G"
K :  c2
K :  "/"
K :  "CO"
K :  "J"
K :
K :  "/"
K :  "MA"
K :  "STO"
K :  c2
K :  "Y"
K :  "/"
K :  "OR"
K :  "R"
K :  "K"
K :  0,a0,0
K :  0,0,2*360/n2
K :  "/"
K :  "MA"
K :  "G"
K :  c2
K :  "/"
K :  "CO"
K :  "J"
K :
K :  "P"
```

```
K : "P"
K : "K"
K : 0,0,-b2/2
K : "D"
K : 0,0,1
K : "P"
C :
K : "P"
K : "P"
K : "K"
K : 0,0,b2/2
K : "D"
K : 0,0,1
K : "N"
C :
K : "P"
K : "P"
K : "K"
K : 0,aa+100,0
K : "D"
K : 0,1,0
K : "N"
K : "/"
K : "MA"
K : "STO"
K : c2
K : "Y"
K : "/"
K : "MO"
K : "DE"
K : "Y"
K : "/"
K : "MA"
K : "PU"
K : c2
K : "Y"
C :
C :
C :        ======================
C :        = creat 3-D worm blank  =
C :        ======================
C :
C :
K : "/"
K : "CR"
K : "DEF"
K : "R"
K : "X"
K : 32
K : "F"
K : "A"
K : 11
K : "D"
K : "D"
K : "REV"
K : p1
K :
K : "/"
K : "OR"
K : "R"
K :
K : 0,0,90
K : "/"
K : "MA"
```

```
K : "STO"
K : w1
K : "G"
K : c2
C :
C :                  =======================
C :                  = 3-D worm cutting =
C :                  =======================
C :
K : "/"
K : "CO"
K : "CU"
K :
K : "/"
K : #time=1
K : #start:
K : #t1=time*dtw
K : #t2=t1/ratio
K : "OR"
K : "R"
K :
K : -dtw,0,0
K : "/"
K : "MA"
K : "G"
K : c2
K : "/"
K : "OR"
K : "R"
K : "K"
K : 0,a0,0
K : 0,0,t2
K : "/"
K : "CO"
K : "CU"
K :
K : "/"
K : #time=time+1
K : #if(time lt wn) then goto start
K : "/"
K : "OR"
K : "R"
K :
K : (time-1)*dt1,0,0
K : "/"
K : "MA"
K : "STO"
K : worm
K :
K : "/"
K : "MO"
K : "DE"
K : "Y"
K : "/"
K : "CR"
K : "REV"
K : p2
K :
K : "/"
K : "OR"
K : "R"
K :
```

```
K : 0,0,90
K : "/"
K : "CO"
K : "J"
K :
K : "/"
K : "MA"
K : "PU"
K : worm
K : "Y"
C :
C :                        ═══════════════
C :                        = 3-D gear cutting  =
C :                        ═══════════════
C :
K : "/"
K : "MA"
K : "G"
K : g2
K : "G"
K : worm
K : "/"
K : "CO"
K : "CU"
K :
K : "/"
K : #time=1
K : #again:
K : #t1=time*dtg
K : #dt2=dtg/ratio
K : "OR"
K : "R"
K : "K"
K : 0,a0,0
K : 0,0,dt2
K : "/"
K : "MA"
K : "G"
K : worm
K : "/"
K : "OR"
K : "R"
K :
K : -t1,0,0
K : "/"
K : "CO"
K : "CU"
K :
K : "/"
K : #time=time+1
K : #if(time lt gn) then goto again
K : "/"
K : "OR"
K : "R"
K : "K"
K : 0,a0,0
K : 0,0,-(time-1)*dt2
K : "/"
K : "MA"
K : "STO"
K : gear
K :
```

```
K : "/"
K : "MO"
K : "DE"
K : "Y"
K : "/"
K : "MA"

K : "PU"


K : gear
K : "Y"
K : #echo all
C :
C :                  ================
C :                  = view worm =
C :                  ================
C :
K : "G"
K : worm
K : "/"
K : "V"
K : "E"
K : 1,1,1,60
K : "MG"
K : "T"
K : "MG"
K : "T"
K : "DO"
K : "HI"
K : "E"
K : "/"
K : "DE"
K : "Y"
C :
C :                  ================
C :                  = view gear =
C :                  ================
C :
K : "/"
K : "MA"
K : "G"
K : gear
K : "/"
K : "V"
K : "RM"
K : 0,0,120
K : "/"
K : "DO"
K : "SH"
K : "E"
C :
E : **** END OF SESSION ****
```

```
c
c     ***************************************************************
c     PROGRAM MAIN:
c
c     This is a part of a developing software package which is used to
c     analyse the double-enveloping wormgear fluid film lubrication.
c     This part includes:
c          (1) calculation of the wormgear dimensions;
c          (2) calculation of the coordinates of worm and gear surfaces
c              in S1 and S2 systems;
c          (3) calculation of the contact lines on all pairs of meshing
c              teeth for a given position;
c          (4) calculation of the coordinates and relative velocity at
c              the worm pitch points on all pairs of meshing teeth for
c              a given position;
c          (5) calculation of load sharing among all meshing teeth for
c              a given position.
c
c
c     To run this program the input parameters are:
c          a0         center distance of the first enveloping      (mm)
c          a2         center distance of wormgear                  (mm)
c          bta        apex angle of the generating plane           (deg)
c          d1         pitch diameter of worm                       (mm)
c          df1        S1 running step size in the first enveloping  (deg)
c          dt2        S2 running step size in the second enveloping (deg)
c          dx         interval along x axis                        (mm)
c          dz         interval along z axis                        (mm)
c          ep,epp     convergence criterion for root searching
c          nl         number of worm thread
c          np         number of meshing teeth
c          nn         number of dimension
c          rfn        normal presure angle                         (deg)
c          ratio      speed ratio of the first enveloping
c          ratio2     speed ratio of wormgear
c          power      output power from gear shaft                 (kw)
c          rpm        input speed from worm shaft                  (rpm)
c          tn2        given gear position                          (deg)
c          uf         friction coefficient
c
c
c     The variables in the main program are:
c          t20 ---- running angle of the second enveloping start   (rad)
c          t21 ---- running angle of the second enveloping end     (rad)
c
c
c     Subroutines included:
c
c     (1) Called by MAIN Program
c
c          DIMEN      develop the dimensions of wormgear.
c          WORM       develop the worm surface in S1.  Here the worm is
c                     ground by a disc grinding wheel.
c          WORML      develop the worm surface on the other side, in S1.
c          GEAR       develop the gear surface in S2.
c          CONTACT    calculate the length of contact lines on each pair
c                     of meshing teeth at the given position.
c          PITCH      finding the pitch points on the worm surface, the
```

101

```
c                   normal directions and relative velocities on each
c                   pair of meshing teeth for a given position.
c        LOAD       calculate the normal force on each pair of meshing
c                   teeth and the torque sharing among the worm meshing
c                   teeth.
c
c    (2) Called by Subroutine
c        EQG        the gear surface equation in S2
c
c        EQW        the worm surface equation in S1
c        EQWL       the worm surface equation of the other side in S1
c        FIRST      calculation of the first enveloping surface
c        NORMAL     calculation of the properties of the worm pitch points
c        ROOT       finding the pitch point
c        SECOND     calculation of the second enveloping surface
c        WORMT      calculation of the worm profile on mid-plane in Sj
c
c
c    Outputs of the program are written in the data files under
c    the names of the subroutines.  Units used in the output data are:
c        rad        for angle
c        mm         for length
c        newton     for force.
c    --------------------------------------------------------------------
c
c        PROGRAM MAIN
c
        implicit double precision(a-h,o-z)
        parameter(bta=5.0,a0=880.00,a2=880.00,ratio=110.,ratio2=110,
     &            rfn=20,n1=1,np=10,d1=300.0,epp=1.0e-5,ep=1.0e-3,
     &            power=4000,rpm=14280,uf=0.03,nn=600)
        dimension contc(np),tp2(np),fp2(np),pxj(np),pzj(np),
     &            xpn(np),ypn(np),zpn(np),vx(np),vy(np),vz(np),
     &            wtor(np),wtorf(np),forn(np),fornf(np),
     &            exaa(nn),exbb(nn),excc(nn),exdd(nn)
        data df1/10.0/,dt2/1.0/,dx/5.0/,dz/5.0/,tn2/0.0/
c
        open(1,file='/dimen dat')
        open(2,file='/worm dat')
        open(3,file='/worml dat')
        open(4,file='/gear dat')
        open(5,file='/contc dat')
        open(7,file='/normal dat')
        open(8,file='/pitch dat')
        open(9,file='/load dat')
c
        pi=4.d0*datan(1.d0)
        bt=bta*pi/180.d0
        tb=dtan(bt)
        dt=dt2*pi/180.d0
        df=df1*pi/180.d0
        rfnr=rfn*pi/180.d0
c
c
c    | calculation of the dimensions |
c
c
        call dimen(a0,a2,ratio2,n1,np,d1,rfnr,bt,
     &            b1,b2,cc,ddd,dlt,f20,f21,
     &            rb,r1,r2,ra1,ra2,rf1,rf2,rmax2,
     &            reps,tt2,cmax,cmin,start,end)
c
        print*,' DIMEN done'
```

```
c
c
c     | calculation of the worm surfaces along the contact line |
c
c
      call worm(a0,a2,ratio,rb,bt,tb,
     &          f20,f21,b1,ra2,rf2,cc,df,dz)

      call worml(a0,a2,ratio,rb,bt,tb,
     &          f20,f21,b1,ra2,rf2,cc,df,dz,reps)
      Print*,'WORM and WORML done'
c
c
c     | calcukation of the gear surfaces along contact lines |
c
c
      t20=f20
      t21=f21
      call gear(a0,a2,ratio,ratio2,rb,tb,bt,rmax2,cmax,cmin,
     &          ra2,rf2,r2,ra1,rf1,r1,cc,ddd,b1,b2,
     &          f20,f21,df,t20,t21,dt,dlt,dz)
      print*,'GEAR done'
c
c
c     | calculation of the contact line at the given position tn2 |
c
c
      call contact(a0,a2,ratio,ratio2,np,rb,tb,bt,dh,ddd,
     &             r1,ra1,rf1,r2,ra2,rf2,rmax2,cmax,cmin,b1,b2,
     &             f20,f21,df,t20,t21,tt2,tn2,start,end,dlt,dz,
     &             contc,ncc,nn,exaa,exbb,excc,exdd)
        print*,'CONTACT done'
c
c
c     | calculate the characters of pitch points on the worm surface |
c     | at the given position 'tn2'                                   |
c
c
      call pitch(a0,a2,ratio,ratio2,np,rb,tb,bt,cc,ddd,
     &           r1,ra1,rf1,r2,ra2,rf2,rmax2,cmax,cmin,b1,b2,
     &           f20,f21,df,t20,t21,tt2,tn2,start,end,dlt,dz,
     &           nn,exaa,exbb,excc,exdd,ep,epp,
     &           tp2,fp2,pxj,pzj,xpn,ypn,zpn,vx,vy,vz,ncp)
        print*,'PITCH done'
c
c
c     | calculate the load distribution at the given position tn2 |
c
c
      if(ncc.ne.ncp) then
        print*, ' ncc.ne.ncp check parameters'
        print*,ncc,ncp
      endif
      call load (a2,ratio2,r2,np,power,rpm,uf,ncc,tp2,pxj,
     &           pzj,xpn,ypn,zpn,contc,wtor,wtorf,forn,fornf)
        print*,'LOAD done'
c
      stop
      end
c
c ***************** END of MAIN PROGRAM *****************************
c
```

103

```
c
c ═══════════════════════════════════════════════════════════════
c    Subroutine DIMEN called by MAIN
c
c    This subroutine is used to develop the dimensions of a wormgear
c
c
c    To run this subroutine the input parameters are:
c
c          a0     center distance of the first enveloping        (mm)
c          a2     center distance of wormgear                    (mm)
c          bta    apex angle of the generating plane             (deg)
c          d1     pitch diameter of worm                         (mm)
c          n1     number of worm thread
c          nbao   number of meshing teeth
c          ratio  speed ratio of wormgear
c          rfn    normal presure angle                           (deg)
c
c
c    The output of the subroutine are:
c          b1     face width of worm                             (mm)
c          b2     face width of gear                             (mm)
c          cmax   maximum throad radius of the gear hob          (mm)
c          cmin   minimum root radius of the gear hob            (mm)
c          dh     clearance                                      (mm)
c          dlt    half of gear face angle                        (rad)
c          dlta   the change in center distance                  (mm)
c          end    the angle of the meshing end                   (rad)
c          f20    the angle of the first enveloping begining     (rad)
c          f21    the angle of the first enveloping ended        (rad)
c          r1     pitch radius of worm                           (mm)
c          r2     pitch radius of gear                           (mm)
c          ra1    throat radius of worm                          (mm)
c          ra2    throat radius of gear                          (mm)
c          rb     radius of base circle                          (mm)
c          rf1    root radius of worm                            (mm)
c          rf2    root radius of gear                            (mm)
c          reps   the angle of grinding wheel turning to another side(rad)
c          rgmax  out-radius of the gear                         (mm)
c          start  the angle of the meshing starting              (rad)
c          tao    angle between two neighboring gear teeth       (rad)
c
c
c    The results which are not called by other subroutines are written
c    in the data file under the name of 'dimen.dat'.
c
c --------------------------------------------------------------------
      subroutine dimen(a0,a2,ratio,n1,nbao,d1,rfn,bta,
     &                 b1,b2,dh,dlta,dlt,f20,f21,
     &                 rb,r1,r2,ra1,ra2,rf1,rf2,rgmax,
     &                 reps,tao,cmax,cmin,start,end)
      implicit double precision(a-h,o-z)
    1 format(2x,6f12.4)
c
c  pitch diameter and radii
c
      n2=idint(n1*ratio)
      d2=2.0*a2-d1
      r1=d1/2.0
      r2=d2/2.0
c
c  circle pitch, thickness, addendum, dedendum and clearance
c
```

```fortran
      pi=4.d0*datan(1.d0)
      px=pi*d2/n2
      s2=0.55*px
      s1=px-s2
      xm=px/pi
c
      gama=datan(px*n1/(pi*d1))
      rfx=dtan(rfn)/dcos(gama)
      rfx=datan(rfx)
      drf=px/(2*d2)
c
      pn=px*dcos(gama)
      ba=.225*pn
      bk=0.45*pn
      bt=pn/2.0
      bf=bt-ba
      dh=bt-bk
c
c   diameters and radius
c
      db=d2*dsin(rfx+drf)
      da1=d1+2.0*ba
      da2=d2+2.0*ba
      df1=d1+2*ba-2*bt
      df2=d2+2*ba-2*bt
      rb=db/2.0
      ra1=da1/2.0
      ra2=da2/2.0
      rf1=df1/2.0
      rf2=df2/2.0
c
      rfi=rfx+drf
      tao=2*pi/n2
      psia=.5*tao*(nbao-.45)
      start=rfi-psia
      end=start+2*psia
      b1=idint(d2*dsin(psia))
      b2=idint(.9*df1)
c
      rgmax=a2-dsqrt(r1*r1-b2*b2/4.0)
      cmax=a2-dsqrt(rf2*rf2-b1*b1/4.0)
      cmin=a2-dsqrt(ra2*ra2-b1*b1/4.0)
      dlta=cmax+rf2-a2
      reps=(2.0*rfi+2*s1/d2)
      f20=start-tao
      f21=end+tao
      dlt=dasin(df1/d1)
c
c   table of the wormgear dimensions
c
      write(1,*) 'Dimensions of selected wormgear'
      write(1,*) 'a0,a2,rfn,ratio,db'
      write(1,1) a0,a2,rfn,ratio,db
      write(1,*) 'p2,s2,s1,bk,bt,cc'
      write(1,1) px,s2,s1,bk,bt,dh
      write(1,*) 'da1,df1,da2,df2,d1,d2'
      write(1,1) da1,df1,da2,df2,d1,d2
      write(1,*) 'ra1,rf1,ra2,rf2,r1,r2'
      write(1,1) ra1,rf1,ra2,rf2,r1,r2
      write(1,*) 'rfi,rfx,drf,reps,lamda,psia'
      write(1,1) rfi,rfx,drf,reps,gama,psia
      write(1,*) 'bta,f20,f21,start,end'
```

```
      write(1,1) bta,f20,f21,start,end
      write(1,*) 'dlt,b1,b2'
      write(1,1) dlt,b1,b2
      return
      end
c
c ============== End of Subroutine DIMEN ==============

C

c
c ================================================================
c    Subroutine WORM called by MAIN
c
c    This subroutine is used to calculate the worm surface in S1, where
c    worm is ground by a disc. To drive this subroutine, the subroutine
c    DIMEN must be called first.
c
c    To run this subroutine the input parameters are:
c         a0     center distance of the first enveloping      (mm)
c         a2     center distance of wormgear                  (mm)
c         b1     face width of worm                           (mm)
c         bt     apex angle of the generating plane           (rad)
c         cc     clearance                                    (mm)
c         df     step size of f1
c         dz     step size of z1
c         f20    the angle of the first enveloping begining   (rad)
c         f21    the angle of the second enveloping ended     (rad)
c         ra2    throat radius of gear                        (mm)
c         ratio  speed ratio of wormgear
c         rb     radius of base circle                        (mm)
c         rf2    root radius of gear                          (mm)
c         tb     tan(bt)
c
c
c    The variables in WORM are:
c
c         f1     the running parameter in the first enveloping  (rad)
c         f2     f1/ratio
c         z1     the coordinate in worm axis direction
c
c    The results which are not called by other subroutines are written
c    in the data file under the name of 'worm.dat'.
c ----------------------------------------------------------------
c
      subroutine worm(a0,a2,ratio,rb,bt,tb,
     &                f20,f21,b1,ra2,rf2,cc,df,dz)
      implicit double precision(a-h,o-z)
    2 format(2x,6f10.4)
c
c    Title of WORM.dat file
c
      write(2,*) '    The contact lines on the worm surfaces in S1'
      write(2,*) '    ----------------------------------------------'
      write(2,*) '      f1   |    f2   |   z1    |   x1    |   y1    |
     & curve1'
      write(2,*) '    ----------------------------------------------
     &---------'
c
c    precalculation
c
      fi=f20*ratio
      fe=f21*ratio
      m=idnint((fe-fi)/df)+1
```

106

```fortran
      zi=-b1/2.0
      l=idint(b1/dz)
c
c     calculation of the worm surface coordinate in S1
c
      do 20 k=1,1
        z=zi+(k-1)*dz
        do 10 j=1,m
          f1=fi+(j-1)*df
          f2=f1/ratio
          call eqw(f1,f2,a0,ratio,rb,tb,
     &              p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
c
          dw=p*(q2*r1-q1*r2)+q*(p1*r2-p2*r1)+r*(p2*q1-p1*q2)
          if(dw.eq.0)  goto 10
          fz=-(p*(q2*s1-q1*s2)+q*(p1*s2-p2*s1)+s*(p2*q1-p1*q2))/dw
          fx=-(q*(r2*s1-s2*r1)+r*(q1*s2-q2*s1)+s*(q2*r1-r2*q1))/dw
          fy=(p*(r2*s1-s2*r1)+r*(p1*s2-p2*s1)+s*(p2*r1-r2*p1))/dw
          redge=dsqrt(fx*fx+fy*fy)
c
          dw=p1*q-p*q1
          x=((q1*r-q*r1)*z-q*s1+q1*s)/dw
          y=-((p1*r-r1*p)*z-p*s1+p1*s)/dw
c
c     judge if the point is in the tooth domain
c
          if(z.lt.fz)  goto 10
          rw=dsqrt(x*x+y*y)
          rmax=a2-dsqrt((rf2+cc)**2-z*z)
          rmin=a2-dsqrt((ra2+cc)**2-z*z)
          if(rw.gt.rmax.or.rw.lt.rmin) goto 10
c
c     justify the tooth undercut
c
          if(redge.gt.rw) then
           print*,'the worm surface intersects with the edge curve'
           print*,x,y,z
          endif
c
c     calculation of the worm surface curvature
c
          ff=p2*x+q2*y+r2*z+s2
          rr=r*r
          pe1=1.0+p*p/rr
          pf1=p*q/rr
          pg1=1.0+q*q/rr
          dw=ff*rr*dsqrt(p*p+q*q+rr)
          aa=p*r1-r*p1
          bb=q*r1-r*q1
          pl1=aa*aa/dw
          pm1=aa*bb/dw
          pn1=bb*bb/dw
          curve1=(pe1*pn1-2.0*pf1*pm1+pg1*pl1)/(pe1*pg1-pf1*pf1)
c
          write(2,2) f1,f2,z,x,y,curve1
10      continue
20    continue
      return
      end
c
c     =============== End of Subroutine WORM ===============
c
```

```
c
c  =============================================================
c     Subroutine EQW called by WORM
c
c     This subroutine is used to calculate the coefficients of the worm
c     surface equations.
c
c     To run this subroutine the input parameters are:
c
c     a0      center distance of the first enveloping      (mm)
c     bt      apex angle of the generating plane           (rad)
c     f1      running parameter in the first enveloping    (rad)
c     f2      f1/ratio
c     rb      radius of base circle                        (mm)
c     tb      tan(bt)
c
c     The output in EQW are:
c
c     p,q,r,s        coefficients of the first equation in Eqs(19)
c     p1,q1,r1,s1    coefficients of the second equation in Eqs(19)
c     p2,q2,r2,s2    coefficients of the third equation in Eqs(19)
c
c  -------------------------------------------------------------
c
      subroutine eqw(f1,f2,a0,ratio,rb,tb,
     &               p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
      implicit double precision(a-h,o-z)
c
         cf2=dcos(f2)
         sf2=dsin(f2)
         cf1=dcos(f1)
         sf1=dsin(f1)
c
c     Here is the equation (18)
c
         p=cf1*sf2+tb*sf1
         q=tb*cf1-sf1*sf2
         r=-cf2
         s=rb-a0*sf2
         p1=tb*cf1*ratio-sf1*sf2*ratio+cf1*cf2
         q1=-tb*sf1*ratio-sf1*cf2-cf1*sf2*ratio
         r1=sf2
         s1=-a0*cf2
         p2=-tb*sf1*ratio*ratio-cf1*sf2*ratio*ratio
     &      -cf1*sf2-2*sf1*cf2*ratio
         q2=-tb*cf1*ratio*ratio+sf1*sf2*ratio*ratio
     &      +sf1*sf2-2*cf1*cf2*ratio
         r2=cf2
         s2=a0*sf2
      return
      end
c
c  ===========END of Subroutine EQW ============================
c
c
c  =============================================================
c     Subroutine WORML called by MAIN
c
c     This subroutine is used to calculate the worm surface in S1, where
c     worm is ground by a disc. To drive this subroutine, the subroutine
c     DIMEN must be called first.
c
c     To run this subroutine the input parameters are:
```

```
c        a0     center distance of the first enveloping     (mm)
c        a2     center distance of wormgear                 (mm)
c        b1     face width of worm                          (mm)
c        bt     apex angle of the generating plane          (rad)
c        cc     clearance                                   (mm)
c        df     step size of f1
c        dz     step size of z1
c        f20    the angle of the first enveloping begining  (rad)
c        f21    the angle of the second enveloping ended    (rad)
c        ra2    throat radius of gear                       (mm)
c        ratio  speed ratio of wormgear
c        rb     radius of base circle                       (mm)
c        reps   angle of the other side of the worm surface (rad)
c        rf2    root radius of gear                         (mm)
c        tb     tan(bt)
c
c
c     The variables in WORML are:
c
c        f1     the running parameter in the first enveloping  (rad)
c        f2     f1/ratio
c        z1     the coordinate in worm axis direction
c
c     The other results which are not called by other subroutines are
c     written in the data file under the name of 'worml.dat'
c     -------------------------------------------------------------------
c
      subroutine worml(a0,a2,ratio,rb,bt,tb,
     &                 f20,f21,b1,ra2,rf2,cc,df,dz,reps)
      implicit double precision(a-h,o-z)
  3   format(2x,6f10.4)
c
c     Title of worml.dat file
c
      write(3,*) '   The contact lines on the other side worm surfaces'
      write(3,*) '   -------------------------------------------------'
      write(3,*) '     f1   |    f2    |   z1    |   x1    |   y1|
     & curve1'
      write(3,*) '   -------------------------------------------------
     &--------'
c
c     precalculation
c
      fi=f20*ratio
      fe=f21*ratio
      m=idint((fe-fi)/df)+1
      zi=-b1/2.0
      l=idint(b1/dz)
c
c     calculation of the worm surface coordinate in S1
c
      do 20 i=1,1
        z=zi+(i-1)*dz
        do 10 j=1,m
          f1=fi+(j-1)*df
          f2=f1/ratio
          f2=f2-reps
c
          call eqwl(f1,f2,a0,ratio,rb,tb,
     &              p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
          dw=p*(q2*r1-q1*r2)+q*(p1*r2-p2*r1)+r*(p2*q1-p1*q2)
          if (dw.eq.0) goto 10
```

109

```fortran
c
          fz=-(p*(q2*s1-q1*s2)+q*(p1*s2-p2*s1)+s*(p2*q1-p1*q2))/dw
          fx=-(q*(r2*s1-s2*r1)+r*(q1*s2-q2*s1)+s*(q2*r1-r2*q1))/dw
          fy=(p*(r2*s1-s2*r1)+r*(p1*s2-p2*s1)+s*(p2*r1-r2*p1))/dw
          redge=dsqrt(fx*fx+fy*fy)
c
c
```

c   judge if the point is in the tooth domain

```fortran
c
          if(z.gt.fz) goto 10
          dw=p1*q-p*q1
          x=((q1*r-q*r1)*z-q*s1+q1*s)/dw
          y=-((p1*r-r1*p)*z-p*s1+p1*s)/dw
          rw=dsqrt(x*x+y*y)
          rmax=a2-dsqrt((rf2+cc)**2-z*z)
          rmin=a2-dsqrt((ra2+cc)**2-z*z)
          if(rw.gt.rmax.or.rw.lt.rmin) goto 10
c
c   judge the tooth undercut
c
          if(redge.gt.rw) then
          print*,'this point of worm surface is on the edge curve'
          print*,x,y,z
          endif
c
c   calculation of the worm surface curverature
c
          ff=p2*x+q2*y+r2*z+s2
          rr=r*r
          pe1=1.0+p*p/rr
          pf1=p*q/rr
          pg1=1.0+q*q/rr
          dw=ff*rr*dsqrt(p*p+q*q+rr)
          aa=p*r1-r*p1
          bb=q*r1-r*q1
          pl1=aa*aa/dw
          pm1=aa*bb/dw
          pn1=bb*bb/dw
          curve1=(pe1*pn1-2.0*pf1*pm1+pg1*pl1)/(pe1*pg1-pf1*pf1)
c
          write(3,3) f1,f2+reps,z,x,y,curve1
   10     continue
   20   continue
        return
        end
c
c =============== End of Subroutine WORML ===============
c
c
c
c =======================================================
c   Subroutine EQWL called by WORML
c
c   This subroutine is used to calculate the coefficients of the worm
c   surface.
c
c   To run this subroutine the input parameters are:
c       a0    center distance of the first enveloping        (mm)
c       bt    apex angle of the generating plane             (rad)
c       f1    running parameter of the first enveloping      (rad)
c       f2    f1/ratio
c       rb    radius of base circle                          (mm)
c       tb    tan(bt)
```

110

```
c
c     The output in EQW are:
c
c         p,q,r,s        coefficients of the worm equation
c         p1,q1,r1,s1    coefficients of the worm equation
c         p2,q2,r2,s2    coefficients of the worm equation
c
c --------------------------------------------------------------------
c
      subroutine eqwl(f1,f2,a0,ratio,rb,tb,
     &                p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
      implicit double precision(a-h,o-z)
c
         cf2=dcos(f2)
         sf2=dsin(f2)
         cf1=dcos(f1)
         sf1=dsin(f1)
c
         p=cf1*sf2+tb*sf1
         q=tb*cf1-sf1*sf2
         r=-cf2
         s=-rb-a0*sf2
         p1=tb*cf1*ratio-sf1*sf2*ratio+cf1*cf2
         q1=-tb*sf1*ratio-sf1*cf2-cf1*sf2*ratio
         r1=sf2
         s1=-a0*cf2
         p2=-tb*sf1*ratio*ratio-cf1*sf2*ratio*ratio
     &      -cf1*sf2-2*sf1*cf2*ratio
         q2=-tb*cf1*ratio*ratio+sf1*sf2*ratio*ratio
     &      +sf1*sf2-2*cf1*cf2*ratio
         r2=cf2
         s2=a0*sf2
      return
      end
c
c =============== End of Subroutine EQWL ===============
c
c
c
c ========================================================
c     Subroutine GEAR called by MAIN
c     This subroutine is used to calculate the gear surface in S2, where
c     the gear hob is ground by a disc. To drive this subroutine, the
c     subroutine DIMEN must be called first.
c
c     To run this subroutine the input parameters are:
c         a0       center distance of the first enveloping      (mm)
c         a2       center distance of wormgear                  (mm)
c         b1       face width of worm                           (mm)
c         b2       face width of gear                           (mm)
c         bt       apex angle of the generating plane           (rad)
c         cc       clearance                                    (mm)
c         cmax     maximum throad radius of the gear hob        (mm)
c         cmin     minimum root radius of the gear hob          (mm)
c         ddd      the change in center distance                (mm)
c         dlt      half of gear face angle                      (rad)
c         df       step size of f1
c         dz       step size of z2
c         f20      the angle of the first enveloping begining   (rad)
c         f21      the angle of the second enveloping end       (rad)
c         r1       pitch radius of worm                         (mm)
c         r2       pitch radius of gear                         (mm)
c         ra1      throat radius of worm                        (mm)
```

111

```
c       ra2     throat radius of gear                        (mm)
c       ratio   speed ratio of wormgear
c       ratio2  speed ratio of wormgear
c       rb      radius of base circle                        (mm)
c       rf1     root radius of worm                          (mm)
c       rf2     root radius of gear                          (mm)
c       rmax2   out-radius of the gear                       (mm)
c       t20     running angle of the second enveloping start (rad)
c       t21     running angle of the second enveloping ended (rad)
c       tb      tan(bt)
c
c    The variables in GEAR are:
c
c       t1    the running parameter in the second enveloping  (rad)
c       t2    t1/ratio
c       z2    the coordinate in gear axis direction
c
c    The results which are not called by other subroutines are written
c    in the data file under the name of 'gear.dat'.
c
c -------------------------------------------------------------------
c
        subroutine gear(a0,a2,ratio,ratio2,rb,tb,bt,rmax2,cmax,cmin,
     &                   ra2,rf2,r2,ra1,rf1,r1,cc,ddd,b1,b2,
     &                   f20,f21,df,t20,t21,dt,dlt,dz)
        implicit double precision(a-h,o-z)
     4  format(2x,6f12.4)
c
c    Title of Gear.dat file
c
        write(4,*)' This is the point coordinates on gear surface:( when
     & t1=f1, the point is on the first contact line)'
        write(4,*) '-------------------------------------------------------
     &----------------------'
        write(4,*) '           t1   |    f1   |     x2    |    y2    |
     & z2   |        r'
        write(4,*) '        -------------------------------------------------
     &------------------'
c
        l=idint(b2/dz)+1
        m=idint((t21-t20)/dt)+1
        df1=df/10.0
        n=idint(4.d0*dlt/df1)+1
c
c    precalculation
c
        do 10 i=1,m
          t2=t20+(i-1)*dt
          t1=t2*ratio2
c
c    calculation of the curved part of the gear surface in S2
c
          do 20 j=1,n
            f1=(t1-2.0*dlt)+(j-1)*df1
            f2=f1/ratio
            if(f2.lt.f20.or.f2.gt.f21) goto 20
            call second(t2,f2,a0,a2,ratio,ratio2,rb,tb,bt,
     &                   x1,y1,z1,x2,y2,z2,xn,yn,zn,
     &                   curve1,curve2,curve,dw)
c
c    judge if the point is in the tooth domain
c
```

```
              if(dw.eq.0)      goto 20
              if(curve.lt.0) goto 20
              if(z2.gt.b2/2.or.z2.lt.-b2/2) goto 20
              rg=dsqrt(x2*x2+y2*y2)
              rmin=(a2+ddd)-dsqrt(cmax**2-z2*z2)
              rmax=(a2+ddd)-dsqrt(cmin**2-z2*z2)
              if (rmax.gt.rmax2) then
              rmax=rmax2
              endif
              if(rg.gt.rmax.or.rg.lt.rmin) goto 20
c
              write(4,4)t1,f1,x2,y2,z2,rg
       20     continue

              if(a0.ne.a2.or.ratio.ne.ratio2) goto 10
c
c   calculation of the planar part of the gear surface in S2
c       ( for the case of f1=t1 in the conventional form )
c
              do 30 k=1,1
                 z2=-b2/2+(k-1)*dz
                 f1=t1
                 f2=t2
                 if(f2.lt.f20.or.f2.gt.f21) goto 30
                 call first(t2,f2,a0,a2,ratio,ratio2,rb,tb,bt,z2,
       &                    x1,y1,z1,y2,x2,xn,yn,zn,
       &                    curve1,curve2,curve,dw)
c
c   judge if the point is in the tooth domain
c
                 if(dw.eq.0)      goto 30
                 if(curve.lt.0) goto 30
                 rg=dsqrt(x2*x2+y2*y2)
                 rmin=(a2+ddd)-dsqrt(cmax**2-z2*z2)
                 rmax=(a2+ddd)-dsqrt(cmin**2-z2*z2)
                 if (rmax.gt.rmax2) then
                 rmax=rmax2
                 endif
                 if(rg.gt.rmax.or.rg.lt.rmin) goto 30
c
              write(4,4)t1,f1,x2,y2,z2,rg
       30     continue
c
       10 continue
          return
          end
c
c ============== End of Subroutine GEAR ====================
c
c
c
c ================================================================
c   Subroutine SECOND called by Subroutine GEAR
c
c   This subroutine is used to calculate the coefficiences of the gear
c   surface equations Eq(28) in techanical report.
c
c
c   To run this subroutine the input parameters are:
c       a0      center distance of the first enveloping        (mm)
c       a2      center distance of wormgear                     (mm)
c       btr     apex angle of the generating plane             (rad)
c       f2      running parameter in the first enveloping      (rad)
```

```
c         ratio   speed ratio of wormgear
c         ratio2  speed ratio of wormgear
c         rb      radius of base circle                          (mm)
c         t2      running parameter in the second enveloping    (rad)
c         tb      tan(bt)
c
c
c     The output in SECOND are:
c
c         curve1          worm surface curvature at the contact point
c         curve2          gear surface curvature at the contact point
c         curve           relative curvature at the contact point
c         dw              index ( dw=0 the point is a singular point )
c         x1,y1,z1        contact point coordinates in S1
c         x2,y2,z2        contact point coordinates in S2
c         xn,yn,zn        contact point coordinates in Sn
c     -------------------------------------------------------------------
c
      subroutine second(t2,f2,a0,a2,ratio,ratio2,rb,tb,btr,
     &                   x1,y1,z1,x2,y2,z2,xn,yn,zn,
     &                   curve1,curve2,curve,dw)
      implicit double precision(a-h,o-z)
c
c     calculate the coordinates of the point for given f1, t1
c
      t1=t2*ratio2
      f1=f2*ratio
      call eqg(t1,t2,f1,f2,a0,a2,ratio,ratio2,tb,rb,btr,
     &         p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
      dw=p*(q2*r1-q1*r2)+q*(p1*r2-p2*r1)+r*(p2*q1-p1*q2)
      if (dw.eq.0) goto 10
      x2=-(q*(r2*s1-s2*r1)+r*(q1*s2-q2*s1)+s*(q2*r1-r2*q1))/dw
      y2=(p*(r2*s1-s2*r1)+r*(p1*s2-p2*s1)+s*(p2*r1-r2*p1))/dw
      z2=-(p*(q2*s1-q1*s2)+q*(p1*s2-p2*s1)+s*(p2*q1-p1*q2))/dw
c
      ct1=dcos(t1)
      st1=dsin(t1)
      ct2=dcos(t2)
      st2=dsin(t2)
      cf2=dcos(f2)
      sf2=dsin(f2)
      cf1=dcos(f1)
      sf1=dsin(f1)
      sft=dsin(f1-t1)
      cft=dcos(f1-t1)
c
c     Coordinate transformation
c
      x1=-x2*ct1*ct2+y2*ct1*st2-z2*st1+a2*ct1
      y1=x2*st1*ct2-y2*st1*st2-z2*ct1-a2*st1
      z1=-x2*st2-y2*ct2
      xn=x2*ct2-y2*st2
      yn=x2*st2+y2*ct2
      zn=z2
c
c     curvature of the gear surface
c
      rr=r*r
      pe2=1+p*p/rr
      pf2=p*q/rr
      pg2=1+q*q/rr
```

114

```
c
      b=p*r1-r*p1
      d=q*r1-r*q1
      c=p*r2-r*p2
      e=q*r2-r*q2
c
      rto=ratio*ratio
      rto2=ratio2*ratio2
      ffx=sf2*ct2*cft+tb*ct2*sft+2.0*cf2*ct2*sft/ratio
     &      +(sf2*ct2*cft-cf2*st2)/rto
      ffy=sf2*st2*cft+tb*st2*sft+2.0*cf2*st2*sft/ratio
     &      +(sf2*st2*cft+cf2*ct2)/rto
      ffz=sf2*sft-tb*cft-2.0*cf2*cft/ratio+sf2*sft/rto
      ffd=a2*(sf2*cft+tb*sft+2.0*cf2*sft/ratio+sf2*cft/rto)
     &      -a0*sf2/rto
      ff=ffx*x2-ffy*y2-ffz*z2-ffd
c
      ftx=sf2*st2*sft-tb*st2*cft+ratio2*(sf2*ct2*cft+tb*ct2*sft)
     &     +(sf2*ct2-cf2*st2*cft+ratio2*cf2*ct2*sft)/ratio
      fty=-sf2*ct2*sft+tb*ct2*cft+ratio2*(sf2*st2*cft+tb*st2*sft)
     &     +(sf2*st2+cf2*ct2*cft+ratio2*cf2*st2*sft)/ratio
      ftz=ratio2*(sf2*sft-tb*cft-cf2*cft/ratio)
      ftd=ratio2*a2*(sf2*cft+tb*sft+cf2*sft/ratio)
      ft=-ftx*x2+fty*y2+ftz*z2+ftd
c
      ttx=-p+2.0*ratio2*(sf2*st2*sft-tb*st2*cft)
     &      +rto2*(sf2*ct2*cft+tb*ct2*sft)
      tty=-q+2.0*ratio2*(sf2*ct2*sft-tb*ct2*cft)
     &      -rto2*(sf2*st2*cft+tb*st2*sft)
      ttz=rto2*(tb*cft-sf2*sft)
      ttd=rto2*a2*(sf2*cft+tb*sft)
      tt=ttx*x2+tty*y2+ttz*z2-ttd
c
      g=ff*tt-ft*ft
      cm=dcos(btr)/(g*rr)
      pl2=cm*(b*b*tt-2*b*c*ft+c*c*ff)
      pm2=cm*(b*d*tt-(b*e+c*d)*ft+c*e*ff)
      pn2=cm*(d*d*tt-2*d*e*ft+e*e*ff)
      curve2=(pe2*pn2-2*pf2*pm2+pg2*pl2)/(pe2*pg2-pf2*pf2)
c
c     curvature of the worm surface
c
            pe1=1.0+p*p/rr
            pf1=p*q/rr
            pg1=1.0+q*q/rr
            dw=ff*rr*dsqrt(p*p+q*q+rr)
            aa=p*r1-r*p1
            bb=q*r1-r*q1
            pl1=aa*aa/dw
            pm1=aa*bb/dw
            pn1=bb*bb/dw
            curve1=(pe1*pn1-2.0*pf1*pm1+pg1*pl1)/(pe1*pg1-pf1*pf1)
c
c     relative curvature
c
            curve=curve2-curve1
c
  10    return
        end
c
c ============== End of Subroutine SECOND ==============
c
```

115

```
c
c  ==========================================================
c  Subroutine EQG called by Subroutine FIRST and SECOND
c
c  This subroutine is used to calculate the coefficients of the gear
c  surface equations.
c
```

c **To run this subroutine the input parameters are:**

```
c      a0     center distance of the first enveloping     (mm)
c      a2     center distance of the first enveloping     (mm)
c      bt     apex angle of the generating plane          (rad)
c      f1     running parameter of the first enveloping   (rad)
c      f2     f1/ratio
c      t1     running parameter of the second enveloping  (rad)
c      t2     t1/ratio
c      ratio  speed ratio of wormgear
c      ratio2 speed ratio of wormgear
c      rb     radius of base circle                       (mm)
c      tb     tan(bt)
c
c  The output in EQW are:
c
c      p,q,r,s        coefficients in Eqs(29)
c      p1,q1,r1,s1    coefficients in Eqs(30)
c      p2,q2,r2,s2    coefficients in Eqs(31)
c
c  ---------------------------------------------------------------------
c      subroutine eqg(t1,t2,f1,f2,a0,a2,ratio,ratio2,tb,rb,btr,
c     &               p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
c      implicit double precision(a-h,o-z)
c
c          ct1=dcos(t1)
c          st1=dsin(t1)
c          ct2=dcos(t2)
c          st2=dsin(t2)
c          cf2=dcos(f2)
c          sf2=dsin(f2)
c          cf1=dcos(f1)
c          sf1=dsin(f1)
c          sft=dsin(f1-t1)
c          cft=dcos(f1-t1)
c
c  Here is the Eqs(29)-(30)
c
c          p=-sf2*ct2*cft-tb*ct2*sft+cf2*st2
c          q=sf2*st2*cft+tb*st2*sft+cf2*ct2
c          r=sf2*sft-tb*cft
c          s=a2*(sf2*cft+tb*sft)-a0*sf2+rb
c          p1=sf2*ct2*sft-tb*ct2*cft-(sf2*st2+cf2*ct2*cft)/ratio
c          q1=-sf2*st2*sft+tb*st2*cft-(sf2*ct2-cf2*st2*cft)/ratio
c          r1=sf2*cft+tb*sft+cf2*sft/ratio
c          s1=-a2*(sf2*sft-tb*cft-cf2*cft/ratio)-a0*cf2/ratio
c          p2=q-ratio2*(sf2*ct2*sft-tb*ct2*cft)
c          q2=-p+ratio2*(sf2*st2*sft-tb*st2*cft)
c          r2=-ratio2*(sf2*cft+tb*sft)
c          s2=a2*ratio2*(sf2*sft-tb*cft)
c
c      return
c      end
c
c  ============== End of Subroutine EQG ==============
c
```

116

```
c
c
c  ═══════════════════════════════════════════════════════════════
c    Subroutine FIRST called by GEAR
c
c    This subroutine is used to calculate the coefficients of the gear
c    surface equations and curvatures.
c
c
c    To run this subroutine the input parameters are:
c        a0       center distance of the first enveloping        (mm)
c        a2       center distance of wormgear                    (mm)
c        btr      apex angle of the generating plane             (rad)
c        f2       running parameter in the first enveloping      (rad)
c        ratio    speed ratio of wormgear
c        ratio2   speed ratio of wormgear
c        rb       radius of base circle                          (mm)
c        t2       running parameter in the second enveloping     (rad)
c        tb       tan(bt)
c        z2       coordinate of the point in S2
c
c    The output in FIRST are:
c
c        curve1         worm surface curvature at the contact point
c        curve2         gear surface curvature at the contact point
c        curve          relative curvature at the contact point
c        dw             index ( dw=0 the point is a singularity point )
c        x1,y1,z1       contact point coordinates in S1
c        x2,y2          contact point coordinates in S2
c        xn,yn,zn       contact point coordinates in Sn
c  --------------------------------------------------------------------
c
        subroutine first(t2,f2,a0,a2,ratio,ratio2,rb,tb,btr,z2,
     &                   x1,y1,z1,y2,x2,xn,yn,zn,
     &                   curve1,curve2,curve,dw)
        implicit double precision(a-h,o-z)
c
c    calculation of the coordinates of the point
c    at the given position f1, t1
c
        t1=t2*ratio2
        f1=f2*ratio
        call eqg(t1,t2,f1,f2,a0,a2,ratio,ratio2,tb,rb,btr,
     &           p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
        dw=p1*q-p*q1
        if (dw.eq.0) goto 10
        x2=((q1*r-q*r1)*z2-q*s1+q1*s)/dw
        y2=-((p1*r-p*r1)*z2-p*s1+p1*s)/dw
c
        ct1=dcos(t1)
        st1=dsin(t1)
        ct2=dcos(t2)
        st2=dsin(t2)
        cf2=dcos(f2)
        sf2=dsin(f2)
        cf1=dcos(f1)
        sf1=dsin(f1)
        sft=dsin(f1-t1)
        cft=dcos(f1-t1)
c
c    Coordinate transformation
c
```

```
        x1=-x2*ct1*ct2+y2*ct1*st2-z2*st1+a2*ct1
        y1=x2*st1*ct2-y2*st1*st2-z2*ct1-a2*st1
        z1=-x2*st2-y2*ct2
        xn=x2*ct2-y2*st2
        yn=x2*st2+y2*ct2
        zn=z2
c
```

c   curvature of the gear surface

```
c

        rr=r*r
        pe2=1+p*p/rr
        pf2=p*q/rr
        pg2=1+q*q/rr
c
        b=p*r1-r*p1
        d=q*r1-r*q1
        c=b
        e=d
c
        rto=ratio*ratio
        rto2=ratio2*ratio2
        ffx=sf2*ct2*cft+tb*ct2*sft+2.0*cf2*ct2*sft/ratio
     &      +(sf2*ct2*cft-cf2*st2)/rto
        ffy=sf2*st2*cft+tb*st2*sft+2.0*cf2*st2*sft/ratio
     &      +(sf2*st2*cft+cf2*ct2)/rto
        ffz=sf2*sft-tb*cft-2.0*cf2*cft/ratio+sf2*sft/rto
        ffd=a2*(sf2*cft+tb*sft+2.0*cf2*sft/ratio+sf2*cft/rto)
     &       -a0*sf2/rto
        ff=ffx*x2-ffy*y2-ffz*z2-ffd
c
        ftx=sf2*st2*sft-tb*st2*cft+ratio2*(sf2*ct2*cft+tb*ct2*sft)
     &     +(sf2*ct2-cf2*st2*cft+ratio2*cf2*ct2*sft)/ratio
        fty=-sf2*ct2*sft+tb*ct2*cft+ratio2*(sf2*st2*cft+tb*st2*sft)
     &     +(sf2*st2+cf2*ct2*cft+ratio2*cf2*st2*sft)/ratio
        ftz=ratio2*(sf2*sft-tb*cft-cf2*cft/ratio)
        ftd=ratio2*a2*(sf2*cft+tb*sft+cf2*sft/ratio)
        ft=-ftx*x2+fty*y2+ftz*z2+ftd
c
        ttx=-p+2.0*ratio2*(sf2*st2*sft-tb*st2*cft)
     &      +rto2*(sf2*ct2*cft+tb*ct2*sft)
        tty=-q+2.0*ratio2*(sf2*ct2*sft-tb*ct2*cft)
     &      -rto2*(sf2*st2*cft+tb*st2*sft)
        ttz=rto2*(tb*cft-sf2*sft)
        ttd=rto2*a2*(sf2*cft+tb*sft)
        tt=ttx*x2+tty*y2+ttz*z2-ttd
c
        g=ff*tt-ft*ft
        cm=dcos(btr)/(g*rr)
        p12=0
        pm2=0
        pn2=0
        curve2=(pe2*pn2-2*pf2*pm2+pg2*p12)/(pe2*pg2-pf2*pf2)
c
c   curvature of the worm surface
c
        pe1=1.0+p*p/rr
        pf1=p*q/rr
        pg1=1.0+q*q/rr
        dw=ff*rr*dsqrt(p*p+q*q+rr)
        aa=p*r1-r*p1
        bb=q*r1-r*q1
```

                                118

```
              pll=aa*aa/dw
              pml=aa*bb/dw
              pnl=bb*bb/dw
              curvel=(pel*pnl-2.0*pfl*pml+pgl*pll)/(pel*pgl-pfl*pfl)
c
c     relative curvature
c
              curve=curve2-curvel
c
  10          return
              end
c
c ================END of Subroutine FIRST ================
c
c
c ================================================================
c     Subroutine CONTACT called by MAIN
c
c     This subroutine is used to calculate the real contact lines in Sn
c     for the given position tn2.  To drive this subroutine, the
c     subroutine DIMEN must be called first and the given position must
c     be given previously.
c
c
c     To run this subroutine the input parameters are:
c          a0      center distance of the first enveloping      (mm)
c          a2      center distance of wormgear                  (mm)
c          b1      face width of worm                           (mm)
c          b2      face width of gear                           (mm)
c          btr     apex angle of the generating plane           (rad)
c          dh      clearance                                    (mm)
c          cmax    maximum throad radius of the gear hob        (mm)
c          cmin    minimum root radius of the gear hob          (mm)
c          ddd     the change in center distance                (mm)
c          dlt     half of gear face angle                      (rad)
c          df      step size of f1
c          dz      step size of z2
c          end     the angle of the meshing ended               (rad)
c          f20     the angle of the first enveloping begining    (rad)
c          f21     the angle of the second enveloping ended      (rad)
c          np      number of meshing teeth
c          r1      pitch radius of worm                         (mm)
c          r2      pitch radius of gear                         (mm)
c          ra1     throat radius of worm                        (mm)
c          ra2     throat radius of gear                        (mm)
c          ratio   speed ratio of wormgear
c          ratio2  speed ratio of wormgear
c          rb      radius of base circle                        (mm)
c          rf1     root radius of worm                          (mm)
c          rf2     root radius of gear                          (mm)
c          rmax2   outradius of the gear                        (mm)
c          start   the angle of the meshing starting            (rad)
c          t20     running angle of the second enveloping start (rad)
c          t21     running angle of the second enveloping ended (rad)
c          tb      tan(bt)
c          tn2     the given position
c          tt2     angle between two neighboring gear teeth     (rad)
c
c
c     The output of the subroutine are:
c          contc   the length of contact lines of each meshing tooth
c          nc      the number of real meshing pair
```

119

```
c
c
c     The variables in CONTACT are:
c          t1    the running parameter in the second enveloping   (rad)
c          t2    t1/ratio
c          z2    the coordinate in gear axis direction
c
C
C
c     The common exchange dimensions are:
c          x,y,z    save the coordinates of contact point on meshing pair
c          nn       maximum dimension number of the exchange dimension
c          rg       save the radius of contact point to gear axis
c
c
c     The results which are not called by other subroutines are written
c     in the data file under the name of 'contc.dat'.
c     -------------------------------------------------------------------
c
        subroutine contact(a0,a2,ratio,ratio2,np,rb,tb,btr,dh,ddd,
     &                   r1,ra1,rf1,r2,ra2,rf2,rmax2,cmax,cmin,b1,b2,
     &                   f20,f21,df,t20,t21,tt2,tn2,start,end,dlt,dz,
     &                   contc,nc,nn,x,y,z,rg)
        implicit double precision(a-h,o-z)
        dimension x(nn),y(nn),z(nn),rg(nn),contc(np)
    5   format(2x,6f10.4)
c
c     Title of contc.dat file
c
        write(5,*)'  The length of contact lines on each meshing teeth'
        write(5,*) ' -------------------------------------------------'
        write(5,*) '     tn1  |  first  |  second  |  total  '
        write(5,*) '     -------------------------------------'
c
c     precalculation
c
        pi=4.d0*datan(1.d0)
        df1=df/100.0
        dz2=dz/10.0
        nc=0
        l=idint(b2/dz2)+1
        m=idint(f21/tt2)+1
        n=idint(4.0*dlt/df1)+1
c
c
c calculation of the coordinates of the contact point
c
        do 10 i=1,m
           t2=tn2+(i-1)*tt2
           if(t2.lt.start.or.t2.gt.end) goto 10
           t1=ratio2*t2
           ct1=dcos(t1)
           st1=dsin(t1)
           ct2=dcos(t2)
           st2=dsin(t2)
           ind=0
c
           do 20 j=1,n
             f1=(t1-2.0*dlt)+(j-1)*df1
             f2=f1/ratio
             if(f2.lt.start.or.f2.gt.end) goto 20
             call second(t2,f2,a0,a2,ratio,ratio2,rb,tb,bt,
     &                  x1,y1,z1,x2,y2,z2,xn,yn,zn,
```

120

```
      &                         curve1,curve2,curve,dw)
c
c     judge if the point is in the tooth domain
c
         if(dw.eq.0)     goto 20
         if(curve.lt.0) goto 20
c
         if(z2.lt.-b2/2.0.or.z2.gt.b2/2.0) goto 20
         rg2=dsqrt(x2*x2+y2*y2)
         rn2=(a2+ddd)-dsqrt((cmax-dh)**2-z2*z2)
         rm2=(a2+ddd)-dsqrt(cmin**2-z2*z2)
         if (rm2.gt.rmax2) then
         rm2=rmax2
         endif
         if(rg2.gt.rm2.or.rg2.lt.rn2) goto 20
c
         if(z1.gt.b1/2.0.or.z1.lt.-b1/2.0) goto 20
         rw1=dsqrt(x1*x1+y1*y1)
         rn1=a2-dsqrt((ra2+dh)**2-z1*z1)
         rm1=a2-dsqrt((rf2+dh)**2-z1*z1)
         if(rw1.gt.rm1.or.rw1.lt.rn1) goto 20
c
         ind=ind+1
         if(ind.gt.nn) then
         print*,' demensions nn is not big enough for contact'
         print*,' increase nn or df in main program. '
         print*,ind
         pause
         endif
         x(ind)=xn
         y(ind)=yn
         z(ind)=z2
         rg(ind)=rg2
20       continue
c
c     calculation of the length of the second contact line
c
         if (ind.ne.0) then
            rcl2=dsqrt((x(ind)-x(1))**2+(y(ind)-y(1))**2
      &                +(z(ind)-z(1))**2)
         endif
         do 30 j=1,ind
            x(j)=0
            y(j)=0
            z(j)=0
30       continue
c
         if(a0.ne.a2.or.ratio.ne.ratio2) goto 10
c
c calculation of the coordinates of the contact point
c    (this is for the situation where f1=t1)
c
         ind=0
         do 60 k=1,1
            z2=-b2/2.0+(k-1)*dz2
            f1=t1
            f2=t2
            if(f2.lt.f20.or.f2.gt.f21) goto 60
c
            call first(t2,f2,a0,a2,ratio,ratio2,rb,tb,bt,z2,
      &                x1,y1,z1,y2,x2,xn,yn,zn,
      &                curve1,curve2,curve,dw)
```

121

```
c
c   judge if the point is in the tooth domain
c
        if(dw.eq.0)     goto 60
        if(curve.lt.0) goto 60
c
        if(z1.gt.b1/2.0.or.z1.lt.-b1/2.0) goto 60
        rw1=dsqrt(x1*x1+y1*y1)
        rm1=a2-dsqrt((rf2+dh)*(rf2+dh)-z1*z1)
        rn1=a2-dsqrt((ra2+dh)*(ra2+dh)-z1*z1)
        if(rw1.gt.rm1.or.rw1.lt.rn1) goto 60
c
        if(z2.lt.-b2/2.0.or.z2.gt.b2/2.0) goto 60
        rg2=dsqrt(x2*x2+y2*y2)
        rn2=(a0+ddd)-dsqrt((cmax-dh)**2-z2*z2)
        rm2=(a0+ddd)-dsqrt(cmin**2-z2*z2)
        if (rm2.gt.rmax2) then
        rm2=rmax2
        endif
        if(rg2.gt.rm2.or.rg2.lt.rn2) goto 60
c
        ind=ind+1
        if(ind.gt.nn) then
        print*,' demensions nn is not big enough for contact'
        print*,' increase nn or dz in main program. '
        print*,ind
        pause
        endif
        x(ind)=xn
        y(ind)=yn
        z(ind)=z2
        rg(ind)=rg2
  60    continue
c
c   calculation of the length of the second contact line
c
        if (ind.ne.0) then
        rcl1=dsqrt((x(ind)-x(1))**2+(y(ind)-y(1))**2
     &          +(z(ind)-z(1))**2)
        endif
c
c   calculation of the sum of two contact lines on the meshing tooth
c
        if(rcl1.eq.0.and.rcl2.eq.0) goto 10
        nc=nc+1
        contc(nc)=rcl1+rcl2
        write(5,5) t1,rcl1,rcl2,contc(nc)
        rcl1=0
        rcl2=0
c
        do 70 j=1,ind
           x(j)=0
           y(j)=0
           z(j)=0
  70    continue
c
  10 continue
c
     return
     end
c
c ==============END of Subroutine CONTACT ================
```

122

```
c
c
c ==================================================================
c    Subroutine PITCH called by MAIN
c
c    This subroutine is used to calculate the coordinates of the worm
c    pitch points, its normal direction and relative velocity at this
c    point in Sj.  All the calculations are for the given position tn2
c    To drive this subroutine DIMEN must be called first and the given
c    position must be input previously.
c
c
c    To run this subroutine the input parameters are:
c         a0      center distance of the first enveloping        (mm)
c         a2      center distance of wormgear                     (mm)
c         b1      face width of worm                              (mm)
c         b2      face width of gear                              (mm)
c         btr     apex angle of the generating plane            (rad)
c         dh      clearance                                       (mm)
c         cmax    maximum throat radius of the gear hob           (mm)
c         cmin    minimum root radius of the gear hob             (mm)
c         ddd     the change in center distance                  (mm)
c         dlt     half of gear face angle                       (rad)
c         df      step size of f1
c         dz      step size of z2
c         end     the angle of the meshing ended               (rad)
c         ep,epp  convergence criterion for root searching
c         f20     the angle of the first enveloping begining   (rad)
c         f21     the angle of the second enveloping ended      (rad)
c         np      number of meshing teeth
c         r1      pitch radius of worm                            (mm)
c         r2      pitch radius of gear                            (mm)
c         ra1     throat radius of worm                           (mm)
c         ra2     throat radius of gear                           (mm)
c         ratio   speed ratio of wormgear
c         ratio2  speed ratio of wormgear
c         rb      radius of base circle                          (mm)
c         rf1     root radius of worm                             (mm)
c         rf2     root radius of gear                             (mm)
c         rmax2   outradius of the gear                           (mm)
c         start   the angle of the meshing starting            (rad)
c         t20     running angle of the second enveloping start (rad)
c         t21     running angle of the second enveloping ended (rad)
c         tb      tan(bt)
c         tn2     the given position
c         tt2     angle between two neighboring gear teeth      (rad)
c
c
c    The output of the subroutine are:
c         nc              the number of real meshing teeth
c         tp2             the position of each meshing pair
c         fp2             the f2 value of pitch points
c         pxj,pzj         the coordinates of pitch points in Sj
c         xpn,ypn,zpn     the normal direction components of
c                         pitch point in Sj
c         vx,vy,vz        the relative velocity components of
c                         pitch point in Sj
c
c
c    The variables in PITCH are:
c         t1    the running parameter in the second enveloping  (rad)
c         t2    t1/ratio
```

123

```
c          z1    the coordinate in worm axis direction
c
c
c    The common exchange dimensions are:
c          ff     save the fl values of the worm pitch points
c          xm,zm  save the coordinates of the worm mid-plane profile
c          rr     save the radius of worm pitch points
c          nn     maximum dimension number of the exchange dimension


c
c
c    The results which are not called by other subroutines are written
c    in the data file under the name of 'pitch.dat'.
c  --------------------------------------------------------------------
c
       subroutine pitch(a0,a2,ratio,ratio2,np,rb,tb,btr,dh,ddd,
     &                  r1,ra1,rf1,r2,ra2,rf2,rmax2,cmax,cmin,b1,b2,
     &                  f20,f21,df,t20,t21,tt2,tn2,start,end,dlt,dz,
     &                  nn,xm,zm,ff,rr,ep,epp,
     &                  tp2,fp2,pxj,pzj,xpn,ypn,zpn,vx,vy,vz,nc)
       implicit double precision(a-h,o-z)
       dimension xm(nn),zm(nn),ff(nn),rr(nn),tp2(np),fp2(np),
     &           pxj(np),pzj(np),xpn(np),ypn(np),zpn(np),
     &           vx(np),vy(np),vz(np)
   6   format(2x,6f12.4)
c
c
c    Title of contc.dat file
c
       write(7,*) '         pitch point and normal direction '
       write(7,*) '       --------------------------------    '
       write(7,*) '       tnl  |    xj   |    zj   |   xpn   |
     & ypn   |    zpn'
       write(7,*) '       --------------------------------------------
     &------------------'
       write(8,*) '         pitch point and relative velocity direction'
       write(8,*) '       ---------------------------------------------',
       write(8,*) '       tnl  |    xj   |    zj   |   rvx   |
     & rvy   |    rvz'
       write(8,*) '       --------------------------------------------
     &------------------'
c
c    Precalculation
c
       nc=0
       y1=0.0
       pi=4.d0*datan(1.d0)
       m=idint(f21/tt2)+1
       start1=start*ratio
       end1=end*ratio
       df1=df/100.0
       m1=idint(pi/(2.0*df1))+1
c
c    Worm profile on the mid-plane (yj=0)
c
       do 50 k=1,m
       ind=0
       fn2=tn2+(k-1)*tt2
       fn1=fn2*ratio2
       if(fn1.lt.start1.or.fn1.gt.end1) goto 50
c
       do 10 i=1,m1
         fl=fn1-pi/4.0+(i-1)*df1
```

124

```fortran
      call wormt(f1,fn1,y1,a0,ratio,rb,tb,btr,x1,z1,mw)
      if(mw.eq.1) goto 10
      if(x1.lt.0) goto 10
      if(dabs(z1).gt.rb) goto 10
c
      rw1=dsqrt(x1*x1+y1*y1)
      rn1=a0-dsqrt((ra2+dh)*(ra2+dh)-z1*z1)
      rm1=a0-dsqrt((rf2+dh)*(rf2+dh)-z1*z1)
      if(rw1.gt.rm1.or.rw1.lt.rn1) goto 10
c
      ind=ind+1
      if(ind.gt.nn) then
      print*,'the dimension is not big enough in PITCH'
      print*,'increase nn in main'
      print*,ind
      pause
      endif
      xm(ind)=x1
      zm(ind)=z1
      rr(ind)=dsqrt((a0-x1)*(a0-x1)+z1*z1)
      ff(ind)=f1
   10    continue
c
c
c    Find worm pitch points
c
      jt=1
      dr=dabs(rr(1)-r2)
      do 20 j=2,ind
         if(dabs(rr(j)-r2).lt.dr) then
         dr=dabs(rr(j)-r2)
         jt=j
         endif
   20    continue
      if(ff(jt).gt.end1.or.ff(jt).lt.start1) goto 25
c
      fl=ff(jt-1)
      fr=ff(jt+1)
      call root(fl,fr,fn1,y1,a0,ratio,rb,tb,btr,r2,epp,f1,x1,z1)
      if(f1.gt.end1.and.f1.lt.start1) goto 25
      rmin=dsqrt((a0-x1)*(a0-x1)+z1*z1)
c
c
c    Direction of normal line and relative velocity at pitch point
c
      call normal(f1,fn1,x1,y1,z1,a0,ratio,rb,tb,btr,
     &            px,py,pz,rx,ry,rz)
      write(7,6) fn1,x1,z1,px,py,pz
      write(8,6) fn1,x1,z1,rx,ry,rz
c
      nc=nc+1
      tp2(nc)=fn2
      fp2(nc)=f1
      pxj(nc)=x1
      pzj(nc)=z1
      xpn(nc)=px
      ypn(nc)=py
      zpn(nc)=pz
      vx(nc)=rx
      vy(nc)=ry
      vz(nc)=rz
c
```

125

```
   25  do 30 i=1,ind
          xm(i)=0
          zm(i)=0
          rr(i)=0
          ff(i)=0
   30  continue
c
```

50 continue

```
c
       return
       end
c
c ============== END of Subroutine PITCH ==================
c
c
c ========================================================
c     Subroutine WORMT called by Subroutine PITCH and ROOT
c
c     This subroutine is used to calculate the coordinates of the worm
c     surface in Sj specially for yj=0.
c
c     To run this subroutine the input parameters are:
c          a0      center distance of the first enveloping       (mm)
c          btr     apex angle of the generating plane            (rad)
c          ratio   speed ratio of the first enveloping
c          rb      radius of base circle                         (mm)
c          tb      tan(bt)
c          fn1     the given position (tn2*ratio2)                   (rad)
c          ft      running parameter of the first enveloping     (rad)
c
c
c     The output of the subroutine are:
c          xm,zm   the coordinates of the worm profile
c          mw      the index of the worm profile point on edge curve (mw=1)
c ----------------------------------------------------------------------
c
       subroutine wormt(ft,fn1,y,a0,ratio,rb,tb,btr,xm,zm,mw)
       implicit double precision(a-h,o-z)
c
       mw=0
       f2=ft/ratio
       f1=ft-fn1
       call eqw(f1,f2,a0,ratio,rb,tb,
      &          p,q,r,s,p1,q1,r1,s1,p2,q2,r2,s2)
c
       dw=p*(q2*r1-q1*r2)+q*(p1*r2-p2*r1)+r*(p2*q1-p1*q2)
       if(dw.eq.0) goto 10
       fz=-(p*(q2*s1-q1*s2)+q*(p1*s2-p2*s1)+s*(p2*q1-p1*q2))/dw
       dw=p*r1-p1*r
       xm= (y*(q1*r-q*r1)+(s1*r-s*r1))/dw
       zm= (y*(p1*q-p*q1)+(p1*s-p*s1))/dw
       if(zm.lt.fz) goto 10
       goto 20
c
   10 mw=1
c
   20 return
       end
c
c=============== END of Subroutine WORMT ================
c
c
```

126

```
c
c
c==================================================================
c     Subroutine ROOT called by Subroutine PITCH
c
c     This subroutine is used to search the pitch point from the worm
c     profile in Sj specially yj=0.  The method used here is called
c     golden division.
c
c
c     To run this subroutine the input parameters are:
c         a0      center distance of the first enveloping        (mm)
c         btr     apex angle of the generating plane            (rad)
c         r       pitch radius of gear                           (mm)
c         ratio   speed ratio of the first enveloping
c         rb      radius of base circle                          (mm)
c         tb      tan(bt)
c         fnl     the given position (tn2*ratio2)                   (rad)
c         fi,fe   interval of f1 for root searching          (rad)
c         epp     convergence criterion for root searching     (rad)
c
c
c     The output of the subroutine are:
c         x1,z1   the coordinates of the worm pitch point in Sj
c         f1      the value of running parameter at pitch point
c     --------------------------------------------------------------
c
      subroutine root(fi,fe,fnl,y,a0,ratio,rb,tb,btr,r,epp,f1,x1,z1)
      implicit double precision(a-h,o-z)
c
          call wormt(fi,fnl,y,a0,ratio,rb,tb,btr,xm,zm,mw)
          gi=dsqrt((a0-xm)*(a0-xm)+zm*zm)-r
          call wormt(fe,fnl,y,a0,ratio,rb,tb,btr,xm,zm,mw)
          ge=dsqrt((a0-xm)*(a0-xm)+zm*zm)-r
  110     df=fe-fi
          if(df.lt.epp) goto 150
          ft1=fi+.382*df
          ft2=fi+.618*df
          call wormt(ft1,fnl,y,a0,ratio,rb,tb,btr,xm,zm,mw)
          g1=dsqrt((a0-xm)*(a0-xm)+zm*zm)-r
          call wormt(ft2,fnl,y,a0,ratio,rb,tb,btr,xm,zm,mw)
          g2=dsqrt((a0-xm)*(a0-xm)+zm*zm)-r
          if(g1*gi.gt.0) goto 120
          fe=ft1
          ge=g1
          goto 110
  120     if(g2*gi.gt.0) goto 130
          fi=ft1
          fe=ft2
          gi=g1
          ge=g2
          goto 110
  130     fi=ft2
          gi=g2
          goto 110
  150     f1=(fi+fe)/2.0
          call wormt(f1,fnl,y,a0,ratio,rb,tb,btr,x1,z1,mw)
          goto 170
  160     mc=1
  170     return
          end
c============ END of Subroutine ROOT ==========================
```

127

```
c
c
c=================================================================
c    Subroutine NORMAL called by Subroutine PITCH
c
c
c    This subroutine is used to calculate the propertities of the worm
c    pitch point in Sj.
c
c    To run this subroutine the input parameters are:
c        a0     center distance of the first enveloping      (mm)
c        btr    apex angle of the generating plane           (rad)
c        f1     the value of running parameter at pitch point
c        ratio  speed ratio of the first enveloping
c        rb     radius of base circle                        (mm)
c        tb     tan(bt)
c        x1,z1  the coordinates of the worm pitch point in Sj
c
c
c    The output of the subroutine are:
c        wnx,wny,wnz  the normal direction components of
c                     pitch point in Sj
c        xw,yw,zw     the relative velocity components of
c                     pitch point in Sj
c -----------------------------------------------------------------
c
      subroutine normal(f1,fn1,x1,y1,z1,a0,ratio,rb,tb,btr,
     &                  wnx,wny,wnz,xw,yw,zw)
      implicit double precision(a-h,o-z)
c
      f2=f1/ratio
      cf2=dcos(f2)
      sf2=dsin(f2)
      cf1=dcos(f1-fn1)
      sf1=dsin(f1-fn1)
c
      wnx=cf1*sf2+tb*sf1
      wny=tb*cf1-sf1*sf2
      wnz=-cf2
      rx=x1*(1-a0/dsqrt(x1*x1+y1*y1))
      ry=y1*(1-a0/dsqrt(x1*x1+y1*y1))
      rz=z1
      xw=wny*rz-wnz*ry
      yw=wnz*rx-wnx*rz
      zw=wnx*ry-wny*rx
      rlu=dsqrt(xw*xw+yw*yw)
      rld=dsqrt(xw*xw+yw*yw+zw*zw)
      rlead=dacos(rlu/rld)
      return
      end
c
c================= END of Subroutine ROOT =================
c
c
c=================================================================
c    Subroutine LOAD called by MAIN
c
c    This subroutine is used to calculate load sharing among the meshing
c    teeth.  All the calculations are under the given position tn2.  To
c    drive this subroutine the subroutines DIMEN, CONTACT and PITCH must
c    be called first.
c
```

128

```fortran
c
c     To run this subroutine the input parameters are:
c        a2            center distance of wormgear            (mm)
c        nc            the number of real meshing pairs
c        nzmax         number of meshing teeth
c        power         output power from gear shaft           (kw)
c        r2            pitch radius of gear                   (mm)
c        ratio2        speed ratio of wormgear
c        rpm           input speed from worm shaft            (rpm)
c        t2            the position of each meshing pair
c        uf            friction coefficient
c        x1,z1         the coordinates of pitch points in Sj
c        pnx,pny,pnz   the normal direction components of
c                      pitch point in Sj
c        cont          the length of contact lines on each meshing pair
c
c
c     The other results which are not called by other subroutines are
c     written in the data file under the name of 'load.dat'
c     -------------------------------------------------------------
c
      subroutine load (a2,ratio2,r2,nzmax,power,rpm,uf,nc,t2,
     &           x1,z1,pnx,pny,pnz,cont,wtor,wtorf,forn,fornf)
      implicit double precision(a-h,o-z)
      dimension t2(nzmax),cont(nzmax),x1(nzmax),z1(nzmax),
     &          pnx(nzmax),pny(nzmax),pnz(nzmax),
     &          rei(10),refi(10),ww(10),wwf(10)
    9 format(2x,6f12.4)
c
c
      write(9,*) '          normal load and worm torque distribution'
      write(9,*) ' --------------------------------------------------'
      write(9,*) '  pair No. |      normal load      |   worm   | to
     &rque'
      write(9,*) ' ---------------------------------------------------
     &---------'
      write(9,*) '  pair No. |  uf=0          uf<>0  |  uf=0    |
     &uf<>0'
      write(9,*) ' ---------------------------------------------------
     &---------'
c
      pi=4.d0*datan(1.d0)
      grpm=rpm/ratio2
      torqeg=9.550*1.0e6*power/grpm
c
c     calculation of the sum of the contact lines
c
      dd=0.0
      ddf=0.0
      do 20 i=1,nc
        pn=dsqrt(pnx(i)*pnx(i)+pny(i)*pny(i)+pnz(i)*pnz(i))
        xn=dabs(pnx(i)/pn)
        yn=dabs(pny(i)/pn)
        zn=dabs(pnz(i)/pn)
        rei(i)=zn*(a2-x1(i))-xn*z1(i)
        pv=dsqrt(r2*r2+(x1(i)*ratio2)**2)
        refi(i)=uf*r2*r2/pv
        ww(i)=yn
        wwf(i)=yn+uf*x1(i)*ratio2/pv
        dd=dd+rei(i)*cont(i)
        ddf=ddf+(rei(i)-refi(i))*cont(i)
   20 continue
```

129

```
c
c   calculate normal load and worm torque distribution (uf=0)
c
      do 30 j=1,nc
        gfor=torqeg*cont(j)/dd
        wtor=gfor*ww(j)*xl(j)
```

c

c calculation of the normal load and worm torque distribution

```
c   with friction
c
        gforf=torqeg*cont(j)/ddf
        wtorf=gforf*wwf(j)*xl(j)
        write(9,9)  t2(j)*ratio2,gfor,gforf,wtor,wtorf
   30 continue
c
      return
      end
c
c=========== END of Subroutine LOAD ======================
c
```

Appendix 3   The Principal Relative Curvatures

Assume that a surface Σ_1 is a simple surface. At a point Q of Σ_1 there exists a normal **n**. The tangent plane of the surface Σ_1 at Q is perpendicular to the normal **n**. Any plane containing the normal **n** is perpendicular to the tangent plane. This plane is called a normal plane of the surface Σ_1 at point Q. The intersection of the normal plane and the tangent plane is a straight line. The direction of the straight line is called a tangent direction of surface Σ_1 at point Q. Therefore, at a point on Σ_1 there is only one normal direction and only one tangent plane, but there are infinitely many tangent directions and infinitely many normal planes.

At point Q the intersection of surface Σ_1 and a normal plane gives a planar curve, correspondingly the intersection of the normal plane and the tangent plane gives a tangent direction. The curvature of the planar curve at point Q is called the normal curvature of surface Σ_1 in this tangent direction. Let the point Q be given by the position vector r(u,v), where u, v are two independent variables. Then the normal curvature is:

$$k_n = \frac{Ldu^2 + 2Mdudv + Ndv^2}{Edu^2 + 2Fdudv + Gdv^2} \tag{a.1}$$

where $E = r_u^2$, $F = r_u \cdot r_v$, $G = r_v^2$, $L = r_{uu} \cdot n$, $M = r_{uv} \cdot n$, $N = r_{vv} \cdot n$; and **n** is the normal direction given by $r_u \times r_v / |r_u \times r_v|$. If the position of Q is given by r(x,y,z) where z=z(x,y) represents the equation of the surface, then E, F, G, L, M, N can be written as:

$$E = 1 + (\frac{\partial z}{\partial x})^2 \ , \quad F = \frac{\partial z}{\partial x}\frac{\partial z}{\partial y}, \quad G = 1 + (\frac{\partial z}{\partial y})^2$$

$$L = \frac{\frac{\partial^2 z}{\partial x^2}}{DB}, \quad M = \frac{\frac{\partial^2 z}{\partial x \partial y}}{DB}, \quad N = \frac{\frac{\partial^2 z}{\partial y^2}}{DB} \tag{a.2}$$

$$\text{where} \quad DB = \sqrt{1 + (\frac{\partial z}{\partial x})^2 + (\frac{\partial z}{\partial y})^2}$$

The sign of the normal curvature is determined by the normal direction. If the normal direction points to the center of curvature center, the normal curvature is positive, otherwise it is negative.

Along different tangent directions the normal curvatures of a surface are different. The principal curvatures are the extreme values of the normal curvatures. The corresponding tangent directions are called the principal directions (Kreyszig, 1959).

If two surfaces Σ_1 and Σ_2 are in meshing at a contact point Q, the normals of the two surfaces at the contact point coincide. Along a tangent direction let the normal curvatures of the two surfaces be $k_n^{(1)}$ and $k_n^{(2)}$, then in this direction the relative curvature of two surfaces is defined as:

$$K_r = k_n^{(1)} - k_n^{(2)} \qquad (a.3)$$

The principal relative curvatures are the extreme values of the relative curvatures, the corresponding directions are the principal relative directions. If the two surfaces are simple surfaces, there are two principal relative curvatures and two principal relative directions.

If the two surfaces contact along a line, the direction of the contact line is one of the principal relative direction and the corresponding principal relative curvature is zero. The other principal relative direction is perpendicular to the direction of the contact line and the principal relative curvature is (Litvin, 1989; Shen, et al. 1983):

$$K_r = \frac{E_1 N_1 - 2F_1 M_1 + G_1 L_1}{E_1 G_1 - F_1^2} - \frac{E_2 N_2 - 2F_2 M_2 + G_2 L_2}{E_2 G_2 - F_2^2} \qquad (a.4)$$

In the second enveloping process the gear surface, Eqs. 27, is generated by the worm surface, Eqs. 18. In this process the two surfaces are in line contact. Let $P = A_\theta^{(21)} F$, the equations of the gear surface can be written as:

$$P = 0$$
$$D_\psi P = 0 \tag{a.5}$$
$$D_\theta P = 0$$

By solving Eqs. 18, the relations of $z_1(x_1,y_1,\psi(x_1,y_1))$ can be found. A point on the worm surface can then be expressed as $r_1(x_1,y_1,z_1(x_1,y_1))$. Further, by solving Eqs. 27, the relations of $z_2(x_2,y_2)$, $\psi(x_2,y_2)$, $\theta(x_2,y_2)$ can be found. A point on the gear surface can be written as $r_2(x_2,y_2,z_2(x_2,y_2))$. Then, by using Eq. a.4 the principal relative curvature is obtained. In this equation, one should make the following substitutions:

$$E_1 = \frac{F_{x_1}^{\ 2}+F_{z_1}^{\ 2}}{F_{z_1}^{\ 2}}, \qquad F_1 = \frac{F_{x_1}F_{y_1}}{F_{z_1}^{\ 2}}, \qquad G_1 = \frac{F_{y_1}^{\ 2}+F_{z_1}^{\ 2}}{F_{z_1}^{\ 2}} \tag{a.6}$$

$$L_1 = \frac{\left| \begin{matrix} F_{x_1} & F_{z_1} \\ F_{x_1\psi_1} & F_{z_1\psi_1} \end{matrix} \right|^2}{DA}, \quad M_1 = \frac{\left| \begin{matrix} F_{x_1} & F_{z_1} \\ F_{x_1\psi_1} & F_{z_1\psi_1} \end{matrix} \right| \cdot \left| \begin{matrix} F_{y_1} & F_{z_1} \\ F_{y_1\psi_1} & F_{z_1\psi_1} \end{matrix} \right|}{DA}, \quad N_1 = \frac{\left| \begin{matrix} F_{y_1} & F_{z_1} \\ F_{y_1\psi_1} & F_{z_1\psi_1} \end{matrix} \right|^2}{DA} \tag{a.7}$$

$$E_2 = \frac{P_{x_2}^{\ 2}+P_{z_2}^{\ 2}}{P_{z_2}^{\ 2}}, \qquad F_2 = \frac{P_{x_2}P_{y_2}}{P_{z_2}^{\ 2}}, \qquad G_2 = \frac{P_{y_2}^{\ 2}+P_{z_2}^{\ 2}}{P_{z_2}^{\ 2}} \tag{a.8}$$

$$L_2 = m\ (\ b^2\ P_{\theta_2\theta_2} - 2bc\ P_{\psi_1\theta_2} + c^2\ P_{\psi_1\psi_1})$$
$$M_2 = m\ (\ bd\ P_{\theta_2\theta_2} - (be + cd\ )\,P_{\psi_1\theta_2} + ce\ P_{\psi_1\psi_1}) \tag{a.9}$$
$$N_2 = m\ (d^2 P_{\theta_2\theta_2} - 2de\ P_{\psi_1\theta_2} + e^2\ P_{\psi_1\psi_1})$$

where

$$DA = F_{z_1}^2\ F_{\psi_1\psi_1}\ \sqrt{F_{x_1}^2 + F_{y_1}^2 + F_{z_1}^2} \tag{a.10}$$

$$b = \begin{vmatrix} P_{x_2} & P_{z_2} \\ P_{x_2\psi_1} & P_{z_2\psi_1} \end{vmatrix} , \qquad c = \begin{vmatrix} P_{x_2} & P_{z_2} \\ P_{x_2\theta_2} & P_{z_2\theta_2} \end{vmatrix}$$

$$d = \begin{vmatrix} P_{y_2} & P_{z_2} \\ P_{y_2\psi_1} & P_{z_2\psi_1} \end{vmatrix} , \qquad e = \begin{vmatrix} P_{y_2} & P_{z_2} \\ P_{y_2\theta_2} & P_{z_2\theta_2} \end{vmatrix} \qquad (a.11)$$

$$g = \begin{vmatrix} P_{\psi_1\psi_1} & P_{\psi_1\theta_2} \\ P_{\psi_1\theta_2} & P_{\theta_2\theta_2} \end{vmatrix} , \qquad m = \frac{\cos\beta}{gP_{z_2}^2}$$

$$F_q = \frac{\partial F}{\partial q} , \qquad F_{qr} = \frac{\partial^2 F}{\partial q \partial r} \qquad (a.12)$$

$$(q, r = x_1, \ y_1, \ z_1, \ \psi_1)$$

$$P_s = \frac{\partial P}{\partial s} , \qquad P_{st} = \frac{\partial^2 P}{\partial s \, \partial t} \qquad (a.13)$$

$$(s, \ t = x_1, \ y_1, \ z_1, \ \psi_1, \ x_2, \ y_2, \ z_2, \ \theta_2)$$

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1994	Final Contractor Report

4. TITLE AND SUBTITLE

Study of the Kinematic and Dynamic Characteristics of a Wormgear Transmission for Helicopter Applications

6. AUTHOR(S)

D.C. Sun and Qin Yuan

5. FUNDING NUMBERS

WU–505–62–0K–00
C–NAG3–1316
1L161102AH45

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

State University of New York at Binghamton
Department of Mechanical and Industrial Engineering
Binghamton, New York 13902–6000

8. PERFORMING ORGANIZATION REPORT NUMBER

E–8633

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Vehicle Propulsion Directorate
U.S. Army Research Laboratory
Cleveland, Ohio 44135–3191
and
NASA Lewis Research Center
Cleveland, Ohio 44135–3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR–195287
ARL–CR–78

11. SUPPLEMENTARY NOTES

Project Manager, David E. Brewe, Materials Division, organization code 5140, NASA Lewis Research Center, (216) 433–6067.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 37

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The first phase of the study of the performance of a wormgear transmission is reported. In this phase the work included the selection of a double-enveloping wormgear type, and its dimensions, suitable for use in helicopter transmissions; the 3-D graphics representation of the selected wormgear using the I-DEAS software; the analysis of the kinematics of meshing; the analysis of load sharing among the meshing teeth; and the implementation of the analyses in a computer program. The report describes the analyses, their results, and the use of the computer programs.

14. SUBJECT TERMS

Wormgear; Gears; Transmission; Kinematic; Dynamic; Helicopter; Gearing

15. NUMBER OF PAGES

139

16. PRICE CODE

A05

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		